


Trabajo de grado

*"Sistema distribuido de tiempo real para la detección de cortes
en la red telefónica"*

*Alumnas: Andrea Artime
Maria Pamela Fernández*

*Director: Armando De Giusti
Co-director: Hugo Ramón*

<p>TES 98/4 DIF-02008 SALA</p>	<p> UNIVERSIDAD NACIONAL DE LA PLATA FACULTAD DE INFORMATICA Biblioteca 50 y 120 La Plata catalogo.info.unlp.edu.ar biblioteca@info.unlp.edu.ar</p> <p> DIF-02008</p>
--	---

Junio 1.998

Capítulo 1

**Objetivo del Problema y
Software a desarrollar**

OBJETIVO DEL TRABAJO

1. Motivación y objetivo general

Se ha definido el desarrollo de un sistema integrado de hardware y software para la detección en tiempo real del corte de cables telefónicos y el comando de las acciones preventivas-correctivas correspondientes. En esta tarea hay un conjunto de alumnos de Ingeniería Electrónica de la U.N.L.P. realizando su trabajo de Proyecto Final con la dirección del Ingeniero Hugo Lorente y se propone el desarrollo de la especificación y el prototipo del sistema de software distribuido.

La descripción abstracta del problema es la siguiente:

Existen un conjunto de centrales telefónicas en la Capital Federal y el Gran Buenos Aires, de las cuales salen conjunto de cables troncales con alrededor de 1000 pares telefónicos cada uno.

Estos cables troncales se dividen sucesivamente en su recorrido, hasta llegar al usuario.

Existe un nodo central de procesamiento que se puede comunicar en tiempo real (en principio por línea telefónica real o dedicada) con cada una de las centrales.

El problema fundamental reside en que al producirse un corte y robo en un cable troncal o en uno de sus derivados (supongamos una rama de 250 pares telefónicos, por ejemplo), el costo de reparación/reposición es muy alto.

Interesa obtener en tiempo real una detección del corte del cable, una estimación de la distancia y recorrido desde la central. Esta información se obtendría en cada central con un equipamiento dedicado provisto por el grupo de Electrónica y con Software.

Inmediatamente de producido el evento de corte de un cable, se interrumpe al procesador principal, donde reside un Sistema de Información Geográfica que dispone de los planos de la zona y tiene una Base de Datos con el recorrido del cableado.

El procesamiento en el nodo principal permite disparar una alarma en tiempo real a la dependencia de seguridad de la zona y prevenir-reprimir el robo. **El tiempo de respuesta global no debe exceder los tres minutos.**

2. Software a desarrollar

Se propone especificar totalmente el sistema de software, a partir de la especificación de hardware establecida por el grupo de Ingeniería Electrónica.

Se simulará el comportamiento temporal de un sistema con una red de Petri, para simular los tiempos de respuesta requeridos.

Se desarrollará el prototipo del software del sistema. Se realizarán corridas tipo a partir de un trace de señales de hardware representativo de los eventos posibles en el mundo real.

Se dispondrá de una demo de un Sistema de Información Geográfica (Agemap 3.5) de la Capital Federal.

Capítulo 2

**Sistemas Distribuidos
de Tiempo Real**

SISTEMAS DE TIEMPO REAL

1. Conceptos sobre sistemas distribuidos

1.1. Complejidad de los sistemas distribuidos.

En el diseño de software para sistemas de tiempo real que no emplean arquitecturas distribuidas, lo más importante es que el sistema operativo sea capaz de soportar características de tiempo real como manejo de interrupciones, asignación de prioridades y administración de procesos eficiente, intervalos de tiempo exactos para la terminación de procesos periódicos, y primitivas para la comunicación sincrónica y asincrónica de procesos. La concurrencia es aparente, los procesos compiten por el uso de un procesador simple. Los otros procesos son suspendidos mientras el proceso asignado ejecuta sus funciones de procesamiento.

El método de diseño de los sistemas de tiempo real no distribuidos es “hardware first”, es decir, la arquitectura de hardware es elegida durante la fase de análisis de sistema.

En cambio, el diseño de sistemas de tiempo real distribuidos es más complejo. El método es “software first”, El software es diseñado para la mejor correspondencia posible con la solución del problema sin considerar la arquitectura de hardware. El hardware distribuido es elegido para proveer el más efectivo soporte para el software. La concurrencia es verdadera, ya que los procesos son mapeados a múltiples procesadores. La concurrencia aparente puede aún estar presente, debido a que no siempre es posible mapear cada proceso a un procesador simple. Un grupo de procesos puede ser asignados a un procesador dado. Estos procesos ejecutarán entonces bajo una concurrencia aparente. Un grupo de procesos comprende un nodo virtual que es mapeado a un nodo físico.

Para ser capaz de seleccionar un conjunto de nodos físicos que soportarán apropiadamente los nodos virtuales de la solución del problema, el diseñador debe poseer un conocimiento más amplio que para sistemas no distribuidos. Estos conocimientos adicionales deben incluir:

- Características de los procesadores.
- Soporte para un entorno de múltiples procesadores.
- Conectividad de los procesadores.
- Comunicación inter-procesador.
- Protocolos de buses.
- Evaluación de performance.
- Tolerancia a fallas.

1.2. Sistemas distribuidos de Tiempo Real.

Un sistema de Tiempo Real puede ser definido como un sistema controlado que ejecuta todas las funciones de los procesos, con un límite de tiempo especificado.

Un sistema de Tiempo Real usualmente incluye un conjunto de dispositivos de hardware independientes que operan a velocidades que difieren ampliamente. Estos dispositivos deben ser controlados de manera que trabajen sincrónicamente. Hay que diseñar sistemas de Tiempo Real que optimicen las capacidades de los dispositivos involucrados, mientras satisfacen los requerimientos de performance establecidos.

Las dificultades de diseño e implementación de sistemas de Tiempo Real son:

- Control de dispositivos de hardware (como líneas de comunicación, terminales y recursos de computadoras)
- Procesamiento de mensajes que llegan a intervalos irregulares, con tasas de entrada fluctuantes y prioridades diferentes.
- Detección y control de condiciones de fallas con facilidades para varios grados de recuperación
 - Manejo de colas y buffers para almacenamiento de mensajes y datos
 - Modelar las condiciones concurrentes en un conjunto adecuado de procesos concurrentes
 - Alocación y control de procesos concurrentes en procesadores
 - Manipular la comunicación y sincronización entre procesos concurrentes para
 - Protección de datos compartidos entre procesos concurrentes
 - Administración y “dispatching” de los procesos concurrentes que compiten por el uso de un recurso
 - Manejo de requerimientos de tiempo rigurosos y especificaciones de performance
 - Testeo y debugging de procesos concurrentes que residen sobre uno o más procesadores
 - Diseño de simuladores para dispositivos de hardware que no estén disponibles para la fase de testeo
 - Reducción de complejidades, transformando los requerimientos en unidades manejables que pueden ser diseñados e implementados por equipos de diseñadores y programadores
 - Selección adecuada de la arquitectura de hardware, que soporte el diseño del software

También se pueden caracterizar los sistemas de Tiempo Real por como operan. Un sistema de Tiempo Real corre en un modo continuo, en un modelo altamente confiable, con detección de errores y recuperación.

Otro aspecto es el procesamiento rápido para satisfacer los requerimientos de performance.

1.3. Tolerancia a fallas.

El modo de ejecución continuo de los sistemas de Tiempo Real implica que estén disponibles y seguros durante largos períodos de tiempo con características de performance aceptables. La disponibilidad puede ser definida como la probabilidad de que un sistema de Tiempo Real operará en un punto del tiempo dado. La seguridad podrá definirse como la probabilidad de que el sistema operará correctamente sobre un período de tiempo especificado.

La performance de los sistemas de Tiempo Real es dependiente del alto grado de correspondencia entre el diseño de software correcto y la especificación de la aplicación, y la arquitectura de hardware apropiada para soportar el diseño. Los componentes de hardware se degradan con el tiempo y eso representa un factor importante en asegurar confiabilidad en los sistemas de Tiempo Real. Es virtualmente imposible probar la correctitud de sistemas amplios y complejos de Tiempo Real, aún después de un extensivo testeo y “debugging” del software. El software no se degrada con el tiempo pero pueden surgir errores de programa. Para satisfacer características de performance de estos sistemas, es necesario incluir un cierto grado de *tolerancia a fallas* desde la primera fase de diseño. La tolerancia a fallas es la habilidad del sistema de continuar la operación durante la presencia de fallas de hardware y errores del software y aún satisfacer al menos una porción de los requerimientos. Un sistema tolerante a fallas debe incluir un mecanismo de detección de errores de software y fallas de hardware, y la corrección de los mismos, mediante un conjunto de procedimientos de recuperación.

1.3.1 Detección y corrección de errores.

La tolerancia a fallas puede ser efectiva solamente si las mismas son detectadas rápidamente. Las fallas de hardware deben ser reparadas dentro de un límite de tiempo, que es único en cada aplicación. Los errores de software deben ser aislados rápidamente y corregidos mediante la restauración de los módulos dañados o de las áreas de datos corrompidas.

Las fallas de hardware típicas son: daños de los circuitos o en las conexiones de los procesadores, módulos de memoria, dispositivos de entrada/salida y links de comunicaciones. Los errores de software más usuales son: datos inconsistentes en las bases de datos; deadlock, inanición y terminación prematura de los procesos; fallas en tiempo de ejecución debido a un valor fuera de rango, por ejemplo división por cero, y falta de almacenamiento para la alocaión dinámica de objetos.

La detección de fallas y errores pueden ser clasificadas dentro de las siguientes categorías:

- Self-Diagnosis.
- Chequeo de errores por otros procesos.
- Chequeo de Software.
- Observación del operador.

1.3.2 Procedimientos de Recuperación.

Sin considerar el método utilizado para detectar una falla, debe ser generada suficiente información para permitir que el procedimiento de recuperación adecuado se ejecute. La información debe incluir al menos la identificación del procesador, nombre del proceso, tipo de error, etc.

El procedimiento de recuperación empleado depende de la aplicación. Numerosos mecanismos son usados para facilitar la tolerancia a fallas en sistemas de Tiempo Real.

1.3.3 Balanceo de carga.

Condiciones de fallas pueden ser causadas por la sobrecarga en uno o más procesadores en el sistema distribuido. Esto podría ser debido a un incremento en el tráfico de mensajes y la inhabilidad de un procesador de ejecutar dentro de las restricciones de tiempo requeridas.

2. Soporte para los sistemas distribuidos

2.1. Especificación de Procesos y Soportes de Comunicación.

2.1.1 Características de Software de Sistemas Distribuidos.

El software de sistemas distribuidos debe proveer herramientas no sólo para el mapeo de componentes de software y programas a los procesadores, sino también para la eficiente comunicación entre procesos. Esto requiere un mecanismo de pasaje de mensajes entre procesos que residen en diferentes procesadores que no comparten memoria y para el manejo de flujos de mensajes cuando el sistema es operacional. El software de sistemas distribuidos de Tiempo Real puede ser caracterizado por el soporte que provee para satisfacer los siguientes requerimientos:

1. Capacidad para especificar módulos de software y caminos de comunicación entre ellos.
2. Un esquema de mapeo de los módulos de software (nodos virtuales) a los procesadores (nodos físicos).

3. Un espacio de nombres para soportar la comunicación entre procesos.
4. Soporte de run-time para comunicación inter e intra-procesos en la forma de priorización, administración y "dispatching" de procesos; manejo de colas de mensajes; manejo de interrupciones; y accesos a los clocks de Tiempo Real.

2.1.2 Modelos de comunicación de procesos.

El mecanismo usado para satisfacer los límites de tiempo de dos procesos que se comunican, y la protección de accesos a los datos compartidos es llamado *sincronización*. La *exclusión mutua* es una abstracción del problema de sincronización. La exclusión mutua permitirá que sólo un proceso acceda a los datos compartidos dentro de una *región crítica*.

El tiempo de comunicación entre los procesos debe ser cuidadosamente controlado y sincronizado.

Los mecanismos para resolver problemas de programación concurrente tradicionales están basados en un conjunto de primitivas, que incluyen instrucciones como fork/join, parbegin/parend, cobegin/coend; semáforos; regiones críticas; monitores y co-rutinas. Es problema del programador que estas primitivas sean usadas correctamente.

En la ingeniería de software moderna el énfasis está en el uso de lenguajes de alto nivel, evitando las primitivas complejas. Dos importantes modelos de concurrencia describen primitivas de programación de alto nivel para usar en la implementación de procesos secuenciales que se comunican. Estos modelos de concurrencia son:

- CSP (Procesos secuenciales de comunicación): especifica como un programa puede ser descrito como un conjunto de procesos independientes que se comunican vía pasaje de mensajes sincrónico. Este modelo no está implementado para sistemas distribuidos,
- DP (Procesos distribuidos): un programa puede ser especificado en términos de un número fijo de procesos concurrentes que son pensados para ser dispersos sobre una red de procesadores distribuidos. No existen estructuras de datos compartidas, las variables son privadas a los procesos que las declaran, y sólo podrán ser accedidas por ellos. La comunicación de procesos es provista mediante llamadas a procedimientos definidos dentro de otros procesos. Los procesos son sincronizados por sentencias no determinísticas llamadas *regiones guardadas*. Estos comandos guardados son provistos por constructores condicionales y loops. La relación llamador-llamado entre procesos no es simétrica, y el proceso llamado no conoce que proceso lo está llamando, como en CSP.

2.1.3. Mecanismos de Comunicación Remota.

Uno de los mecanismos de lenguajes independientes usado para proveer soporte para comunicación inter-procesos es el RPC (Remote Procedure Call). Un proceso *cliente* sobre un procesador envía mensajes hacia o recibe mensajes desde un proceso *servidor* sobre otro procesador. El proceso cliente usa una RPC. Un proceso agente sobre el procesador del servidor recibe la RPC y le asigna un proceso servidor para su ejecución. El proceso servidor ejecuta la RPC y retorna resultados, si hay, al cliente posiblemente vía uno o más agentes intermediarios.

Las RPC pueden ser implementadas con un mecanismo sincrónico o asincrónico, o puede estar restringida a pasaje de mensajes sincrónico. Una implementación sincrónica significa que el proceso cliente ejecuta la RPC y espera por una respuesta del proceso servidor. Una implementación asincrónica implica que el proceso cliente usa la RPC y continúa con su ejecución sin esperar una respuesta del servidor, que podría ser recibida más tarde.

La forma más general para servicios de RPC incluyen soporte para lenguajes de programación y arquitecturas de procesadores heterogéneas.

La construcción de un programa de aplicación y el uso de servicios de una RPC puede ser dividido en tres fases:

- *Tiempo de compilación.* Clientes y servidores son creados como si ellos pudiesen estar linkeados a la misma imagen ejecutable sobre un procesador. Estos clientes y servidores hacen llamadas a clientes y servidores agentes que contienen interfaces a servicios que manejan nombres de clientes y servidores que residen sobre varios procesadores. La existencia de agentes debe ser oculta para los programadores de aplicaciones.
- *Tiempo de ligadura.* Esta fase incluye:
 - Nombramiento.
 - Activación.
 - Determinación del port.
- *Tiempo de llamada.* Los tres componentes que incluye esta fase son:
 - Protocolo de transporte.
 - Protocolo de control.
 - Representación de los datos.

El mecanismo de RPC es usado en sistemas con bajo acoplamiento que no comparten memoria. Los procesadores pueden ser homogéneos o heterogéneos.

Otro modelo para comunicación remota puede ser *multiprocesamiento simétrico*, donde la distribución es no determinística y el programador de la aplicación no tiene control sobre la configuración de la distribución.

2.2. Entorno de soporte para Sistemas distribuidos de Tiempo Real.

El problema encontrado en la creación de un entorno para sistemas distribuidos es significativamente más complejo que para sistemas no distribuidos. Ejemplos de esta complejidad incluyen múltiples generadores de código para procesadores heterogéneos, diferentes representaciones de datos para objetos del mismo tipo, comunicación inter-procesos, y un “debugging” simbólico para múltiples máquinas. El entorno de soporte debe proveer facilidades para la distribución de los módulos de software entre los procesadores identificados para una arquitectura dada. Debería soportar también la reconfiguración de la arquitectura de hardware y la realocación de los módulos de software en caso de fallas de hardware.

2.2.1 Tiempo Real Ejecutivo

El ejecutivo de Tiempo Real debe proveer un amplio rango de características en tiempo de ejecución para soportar las funciones críticas requeridas por los sistemas de Tiempo Real. Estas características incluyen administración, colas y “dispatching” de tareas, manejo de interrupciones, soporte de tiempo exacto, manejo de memoria, manejo de fallas, arbitrar buses, y operaciones multi-procesador. Las operaciones multi-procesador incluyen soporte varios procesadores sobre el mismo bus, múltiples buses, o en redes de área local (LANs), comunicaciones inter-procesadores, y manejo de los recursos del sistema. Algunas de estas características pueden ser implementadas como un *controlador* de funciones sobre procesadores especiales, por ejemplo, un controlador de bus o una unidad de manejo de memoria.

El soporte global para sistemas distribuidos está usualmente limitado para los sistemas que comparten memoria, donde el ejecutivo determina la distribución de los procesos en los procesadores y la administración de los procesos. En este caso la comunicación de los procesos es programada sin tener en cuenta la distribución, y la comunicación inter-procesadores es transparente al programador. El acercamiento más común para los procesadores con bajo acoplamiento (no comparten memoria) es tener un kernel de Tiempo Real residiendo en cada procesador, con los programas de aplicación usando las primitivas de comunicación provistas por el kernel, y el programador debe estar enterado de la configuración de la distribución.

Algunos de los primeros ejecutivos para sistemas de Tiempo Real, fueron implementados en un entorno Unix pero no proveen los servicios requeridos por el hard de los sistemas de Tiempo Real. Ejecutivos especiales de Tiempo Real como VRTX y MTOS fueron desarrollados como kernels con máquinas “desnudas” o en cooperación con otros sistemas operativos. Un *kernel* puede ser descrito como la porción mínima de un sistema operativo que provee servicios especiales para aplicaciones de Tiempo Real. Algunos de estos servicios incluyen:

- Rápida respuesta a eventos asincrónicos.
- Manejo de interrupciones.
- Administración y dispatching de tareas.
- Respuesta a eventos con tiempo.
- Proveer exclusión mutua.

2.2.2 Servicios del ejecutivo.

1. *Administración de procesos*: En la mayoría de los casos habrá múltiples procesos compitiendo por un procesador, lo que requiere la administración y manejo de los mismos.
2. *Puntos de sincronización*: Si el pasaje de mensajes sincrónico o la exclusión mutua para memoria compartida son usados para la comunicación entre procesos, un mecanismo de sincronización debe ser provisto para los elementos concurrentes. El uso de puntos de sincronización implica que al menos un proceso es suspendido durante el período de sincronización. Si esto no es aceptable, un mecanismo diferente como comunicación asincrónica vía colas o mailboxes debe ser provista.
3. *Clocks de Tiempo Real*: Deben ser provistos servicios para iniciar un evento inmediato y medir la expiración del retardo del evento para sistemas con restricciones de tiempo específicas. Debe también ser posible trasladar tiempo real de un procesador a otro. Esto podría involucrar el cálculo del offset relativo entre los tiempos reales de dos procesadores diferentes. Consideraciones especiales deben ser tenidas en cuenta sobre “inmediatos” y “retardo” de eventos para la comunicación inter-procesadores. La resolución del clock del tiempo real es importante, así también como los programadores de aplicaciones pueden especificar eventos con tiempo. Tanto la especificación de tiempo absoluto como de los intervalos de tiempo deberían estar disponibles.
4. *Primitivas de procesos*: Deberían ser provistas primitivas para la creación, activación y terminación de procesos y para la comunicación entre ellos.
5. *Distribución de procesos*: Debe ser posible distribuir un conjunto de procesos entre un conjunto dado de procesadores. Esto podría ser hecho estáticamente con una configuración fija para la duración de la operación del sistema o dinámicamente con o sin balanceo de carga.
6. *Manejo de interrupciones*: Interrupciones para eventos asincrónicos deben ser manejadas eficientemente. Deben ser incluidas características de encolamiento de múltiples

Capítulo 3

Especificación y Diseño del Software

ESPECIFICACIÓN Y DISEÑO DEL SOFTWARE

1. Introducción

1.1. El desafío de la especificación y diseño del software

El desafío más difícil de la ingeniería del software, es el desarrollo del software en el contexto de amplios desarrollos de sistemas de hardware-software combinados. Ejemplos de éstos sistemas son control de tráfico aéreo a nivel nacional, sistemas de comando y control militares, refinerías de aceites automatizadas o plantas de procesamiento de acero. En vías de crecimiento, mecanismos como automóviles, copiadoras e impresoras tienen características similares. Estos sistemas no necesitan ser físicamente amplios; ellos necesitan solamente ser complejos e involucran un número de componentes que no están identificados al comienzo del desarrollo. En otras palabras, el sistema, no solo el software, necesita ser diseñado. Si nosotros desarrollamos procesos y métodos efectivos para dirigir estos sistemas, ellos podrán ser reducidos a desarrollo de sistemas más pequeños o proyectos de sólo software.

Se dispone de un proceso y una metodología para especificación y diseño de software en el contexto de sistemas, como los mencionados anteriormente, que son caracterizados por un desarrollo multi-fase, en los cuales la disciplina de *Ingeniería de Sistemas* juega un papel importante. La Ingeniería de Sistemas es la disciplina que transforma las necesidades del usuario o los requerimientos operacionales en requerimientos de performance del sistema y en un diseño del sistema completo. La Ingeniería de Sistemas también juega un rol importante durante el diseño de hardware y software y coordina especialmente disciplinas de la ingeniería, tales como seguridad, confiabilidad, mantenimiento, disponibilidad, interface hombre-máquina, etc.

Proceso: La figura 1.1 ilustra las fases o actividades en desarrollo de sistemas amplios, con particular énfasis en el desarrollo de hardware y software.

La fase de *Análisis de Requerimientos del Sistema* está dirigida a los requerimientos del sistema completo, la funcionalidad del sistema debe satisfacer las necesidades o requerimientos operacionales del cliente. Esto es el *qué* del sistema.

La fase de *Diseño del Sistema* establece el diseño completo del sistema, tanto los subsistemas de hardware y software como la alocaión de los requerimientos del sistema definidos en la etapa anterior, para los componentes de hardware y software. Esto es el *cómo* del sistema.

La fase de *Análisis de Requerimientos del Software* realiza el análisis de requerimientos de los subsistemas de software que componen el sistema a desarrollar. Define *qué* será hecho por el software de cada subsistema. La *especificación del software* es el producto o documentación resultante de ésta fase.

Las fases del *Diseño del Software* son análogas a las fases del diseño del sistema, en el sentido que definen componentes (piezas del software) que satisfacen los requerimientos: el *cómo* del software. El *Diseño del Software* es el producto o documentación resultante de ésta fase.

Metodología: El software desarrollado en tal contexto se llama *software del sistema*, y aquí son colectivamente referidos como “Metodología de Ingeniería de Software de Sistemas” (SSEM). Los métodos componentes de SSEM, como muestra la figura 1.1 son:

- Análisis estructurado de Tiempo Real (RTSA).

- Diseño de Tiempo Real (RTD)
- Diseño modular estructurado (MSD)

RTSA es el acercamiento hacia el análisis estructurado que incorpora aquellos elementos necesarios para el diseño de software de Tiempo Real.

RTD es el acercamiento para realizar la transición desde los requerimientos del software hacia el diseño de software, finalizando con un diseño completo top-level (arquitectura) del diseño del software. Está desarrollado en suficiente detalle para mostrar exactamente cómo los requerimientos son agrupados y alocados en los componentes del software.

MSD es el método que se usa en la última parte del diseño top-level y finaliza en un diseño detallado y un código resultante. Trata primariamente con sistemas secuenciales o con las partes del software de Tiempo Real que son ejecutadas como un simple thread de control.

2. Ingeniería de Sistemas y el proceso de desarrollo del Sistema

2.1. Desarrollo de grandes sistemas

Proceso y metodología: para proveer un desarrollo ordenado de grandes sistemas, es necesario definir una *metodología*: un conjunto de procedimientos, gráficos y guías que explican como realizar la ingeniería de sistemas y software. Es importante que la metodología esté embebida dentro de un *proceso*: el conjunto de pasos que deben ser realizados para completar una tarea. Un modelo de proceso ejecuta dos funciones primarias:

1. Determina el *orden de las capas* involucradas en el desarrollo y evolución.
2. Establece el *criterio de transición* para progresar desde una capa a la siguiente, el cual consiste del criterio de *completitud* para la fase corriente y el criterio de *entrada* a la fase siguiente.

El modelo de proceso cuestiona al desarrollador del sistema.

1. ¿Qué se deberá hacer después?
2. ¿Cuándo termina de hacerlo?

El énfasis de la metodología está en:

1. *Cómo navegar a través de una fase*: determinando y alocando requerimientos, definiendo arquitectura de sistemas y software, criterios de partición, creación de estructuras de datos y algoritmos.
2. *Cómo representar los productos de una fase*: diagramas de flujos de datos, diagramas de transición de estados, cartas de estructuras, módulos de ocultamiento de información, lenguajes de diseños de programas.

La fase de Análisis de Requerimientos del Sistema es la transformación de las necesidades operacionales del usuario en un conjunto de capacidades y requerimientos de performance adecuados para la determinación interactiva y paralela del diseño del sistema y la alocación de los requerimientos en los componentes del diseño. Las actividades dentro de esta fase son:

- Establecer los objetivos del sistema
- Identificar funciones e interacción entre las funciones
- Especificar performance para cada función

- Definir conceptos operacionales completos

La fase de Diseño del Sistema es la determinación de la arquitectura completa del mismo, que consiste de un conjunto de componentes de procesamiento físicas (hardware, software, personas y la comunicación entre ellas) que satisfacen los requerimientos esenciales del sistema. Las actividades dentro de esta fase son:

- Determinar las alternativas de diseño, incluyendo componentes e interfaces
- Alocar los requerimientos en las componentes del diseño
- Análisis de las alternativas de diseño y el grado de satisfacción de los requerimientos
- Interacción con el análisis de requerimientos
- Determinación y documentación de la alternativa elegida

La fase de Análisis de Requerimientos del Software es la especificación detallada de la funcionalidad necesaria para cada ítem de configuración de software y hardware y de las interfaces entre dichos ítems.

La fase de Diseño Top Level es la determinación y descripción de la arquitectura de software completa, de las interfaces entre los módulos más importantes dentro de la arquitectura y de las interfaces entre los diferentes ítems de configuración de software y hardware. Las actividades dentro de esta fase son:

- Determinar interfaces de hardware
- Asignar tareas a cada interface de hardware
- Descomponer las partes más importantes del software a través de diagramas de flujo
- Determinar concurrencia, identificando tareas dentro de los diagramas de flujo y control
- Encapsular tareas en módulos, usando abstracción de datos y principios de ocultamiento de la información
- Traducir el diseño en gráficos y Lenguaje de Diseño de Programas

La fase de Diseño Detallado es la determinación de la arquitectura y especificación del sistema de nivel mas bajo, en un formato pseudo-código, de cada componente del software.

3. Documentación

- **Ingeniería del Sistema**

Análisis de Requerimientos del Sistema

DCD - Diagrama de Contexto de Datos:

Describe el comportamiento funcional del sistema, resumiendo los requerimientos del sistema a desarrollar, el cual acepta entradas de entidades externas, generando las salidas asociadas a dichas entidades. Estos requerimientos son puramente funcionales e ignoran cualquier diseño o implementación. También son mostradas las interfaces requeridas entre las entidades externas y el sistema.

DFD - Diagrama de Flujo de Datos:

Son el fundamento de la representación gráfica para el análisis de requerimientos del sistema. Son contruidos en niveles, donde cada nivel provee más detalles de los requerimientos. Los DFDs constan de dos partes: los diagramas y la especificación de los procesos.

Diagrama: cada nivel de los DFDs muestran más detalles de los requerimientos y pueden incluir procesos, flujos de datos y almacenamiento de datos. Cada nuevo nivel es una explosión de un proceso mostrado en un nivel más alto.

Pspec - Especificación de Procesos: muestra como la salida para un proceso dado es generada desde su entrada.

Diccionario de Datos:

Consiste de las definiciones de toda la información (datos y control) que fluye a través del sistema.

CCD - Diagrama de Contexto de Control:

Establece el control de la información de las interfaces entre el sistema y su entorno.

CFD - Diagrama de Flujos de Control:

Son imágenes de los DFDs que muestran los flujos de control en vez de datos a través del sistema. Los CFDs constan de dos partes: los diagramas y la especificación de controles.

Diagrama: incluyen los mismos procesos del correspondiente DFD.

Cspec - Especificación de Control: representa el comportamiento del control del sistema. El propósito es mostrar como sus salidas son generadas desde sus entradas. Los tipos de gráficos incluidos son:

Tablas de Decisión

PAT: Tablas de Activación de Procesos

STD: Diagrama de Transición de Estados

STT: Tabla de Transición de Estado

RTS - Especificación de Tiempos de Respuesta:

Es usada como un vehiculo para pensar al sistema como un modelo conducido a eventos. Ilustra los tiempos de respuesta externos que necesitan ser especificados por el modelo de requerimientos. A lo largo del modelo de requerimientos y las características del hardware determinadas durante el diseño del sistema, esta especificación se transforma en una de las mayores entradas para los diseñadores del software.

RTM - Matriz de Traceability de Requerimientos:

Debe proveer asegurar completitud y consistencia con las necesidades del usuario. Puede ser hecha en la forma de una tabla simple que lista las necesidades del usuario a lo largo de un eje, y los componentes del modelo de requerimientos sobre el otro.

EAD - Diagrama Evento-Acción:

El propósito es proveer de un punto de comienzo para el desarrollo del diagrama de flujos de datos. Permite comenzar con un pseudo-DFD basado en estímulos externos (eventos) y respuestas a estos estímulos (acciones). Consta de los siguientes componentes:

Lista de Eventos: una lista de los eventos para el sistema puede ser realizada. Los eventos son las interacciones entre el sistema y su entorno. Los mismos pueden ser temporales o externos. Los eventos externos usualmente producen una entrada de un flujo de datos o de control que cruza los límites del sistema. Un evento temporal arriba en un tiempo particular.

Diagrama : es una forma de DFD/CFD con restricciones especiales. Se dibuja a partir de la lista de eventos, y representa las respuestas a estos eventos.

Matriz o lista de Acción/Almacenamiento: muestra cuáles acciones setean o usan los diferentes almacenamientos. Ayuda a construir el DFD 0 porque agrupa las acciones que operan sobre los mismos almacenamientos.

Diseño del Sistema

Framework de la Arquitectura:

Durante el diseño del sistema, temas específicos del hardware requerirán consideraciones de factores, incluyendo requerimientos adicionales que los componentes del sistema deben conocer, que no están tratados durante la fase de análisis de requerimientos del sistema. Las siguientes son áreas en las cuáles éstas consideraciones surgen:

Procesamiento de la interface de usuarios

Mantenimiento, testeo, y procesamiento de redundancia

Procesamiento de las entradas

Procesamiento de las salidas

ACD - Diagrama de Contexto de Arquitectura:

Diagrama top-level para el modelo de la arquitectura. Contiene la misma información que el DCD y el CCD del modelo de requerimientos, mostrando el lugar del sistema en su entorno y la interface física actual con el entorno.

AFD - Diagrama de Flujo de Arquitectura:

Es una red de representación de la configuración física del sistema. Consiste de dos partes:

Diagrama: documenta los flujos de información entre los módulos de la arquitectura en el sistema. También representa la alocaión de procesos y flujos desde los DFDs y CFDs del modelo de requerimientos en los módulos de la arquitectura.

Mspec - Especificación de los módulos de la Arquitectura: debe ser escrita para cada módulo de arquitectura en el modelo. El propósito es describir la información y procesamiento del módulo en una forma gráfica o narrativa.

AID - Diagrama de la Interconexión de la Arquitectura:

Es la representación de los canales de flujo de información por los cuales se comunican los módulos de la arquitectura. Hay un AID correspondiente con cada AFD. Consiste de:

Diagrama: los canales representan el medio físico por el cual la información viaja desde un módulo de arquitectura a otro. Este medio puede ser material o de energía, como buses eléctricos, links mecánicos o links ópticos.

AIS - Especificación de la Interconexión de la Arquitectura: establece las características del medio físico que conecta los módulos de arquitectura. Estas características pueden ser descriptas en forma textual o gráfica, y deben estar definidas apropiadamente.

Diccionario de la Arquitectura:

Lista todos los elementos de datos y de control entre los módulos de arquitectura y las entidades externas.

RTA - Alocaión de Tiempos de Respuesta:

La especificación de los tiempos de respuesta desde el modelo de requerimientos debe ser alocada en los módulos de arquitectura, vectores de flujo de información y canales de flujo de información en la misma manera como los otros componentes del modelo de requerimientos son alocados. Los requerimientos de tiempo en el modelo de arquitectura debe tener consideración no sólo con las señales externas, sino también con las restricciones de tiempo específicas de los módulos de arquitectura internos.

RTM - Matriz de Traceability de Requerimientos:

Representa la alocaión de los requerimientos del sistema para el diseño de sistema elegido.

- **Ingeniería del Software**

Análisis de Requerimientos del Software

EFD - Diagrama de flujo mejorado:

El propósito es proveer un punto de comienzo para el desarrollo de los diagramas de flujos de datos. Permite comenzar con un DFD que captura todos los requerimientos del sistema (desde el análisis de requerimientos del sistema) que han sido alocados como ítems de software durante el diseño del sistema y entonces agregar todos los requerimientos derivados, basándose en la arquitectura del sistema del diseño del sistema. Estos procesos serán agregados en procesos de un nivel más alto para formar el DFD 0.

DCD - Diagrama de Contexto de Datos:

Describe el comportamiento funcional del sistema, resumiendo los requerimientos del sistema a desarrollar, el cual acepta entradas de entidades externas, generando las salidas asociadas a dichas entidades. Estos requerimientos son puramente funcionales e ignoran cualquier diseño o implementación. También son mostradas las interfaces requeridas entre las entidades externas y el sistema.

DFD - Diagrama de Flujo de Datos:

Son el fundamento de la representación gráfica para el análisis de requerimientos del software. Son contruidos en niveles, donde cada nivel provee más detalles de los requerimientos. Los DFDs constan de dos partes: los diagramas y la especificación de los procesos.

Diagrama: cada nivel de los DFDs muestran más detalles de los requerimientos y pueden incluir procesos, flujos de datos y almacenamiento de datos. Cada nuevo nivel es una explosión de un proceso mostrado en un nivel más alto.

Pspec - Especificación de Procesos: muestra como la salida para un proceso dado es generada desde su entrada.

Diccionario de Datos:

Consiste de las definiciones de toda la información (datos y control) que fluye a través del sistema.

CCD - Diagrama de Contexto de Control:

Establece el control de la información de las interfaces entre el sistema y su entorno.

CFD - Diagrama de Flujos de Control:

Son imágenes de los DFDs que muestran los flujos de control en vez de datos a través del sistema. Los CFDs constan de dos partes: los diagramas y la especificación de controles.

Diagrama: incluyen los mismos procesos del correspondiente DFD.

Cspec - Especificación de Control: representa el comportamiento del control del sistema. El propósito es mostrar como sus salidas son generadas desde sus entradas. Los tipos de gráficos incluidos son:

Tablas de Decisión

PAT: Tablas de Activación de Procesos

STD: Diagrama de Transición de Estados

STT: Tabla de Transición de Estado

RTS - Especificación de Tiempos de Respuesta:

Es usada como un vehículo para pensar al sistema como un modelo conducido a eventos. Ilustra los tiempos de respuesta externos que necesitan ser especificados por el modelo de requerimientos. A lo largo del modelo de requerimientos y las características del hardware determinadas durante el diseño del sistema, esta especificación se transforma en una de las mayores entradas para los diseñadores del software.

RTM - Matriz de Traceability de Requerimientos:

Durante el análisis de requerimientos del software representa la alocaión de los requerimientos del sistema (desde el modelo de requerimientos del sistema y el modelo de la arquitectura del sistema) en componentes de modelo de requerimientos del software.

Diseño del Software

Diseño Top-Level

FFD - Diagrama de Flujos Plano:

El propósito es proveer un punto de comienzo para el desarrollo de la arquitectura del software. Permite comenzar con un DFD y un CFD que capturen todos los requerimientos del software desde el análisis de requerimientos del software, y entonces agregar estos procesos en procesos de un nivel más alto basándose en la concurrencia (dos o más tareas operan simultáneamente) de los items de software.

TCG - Gráfico de Comunicación de Tareas:

Provee una representación gráfica (independiente de un lenguaje) para las tareas concurrentes y las interfaces entre ellas.

SAD - Diagrama de la Arquitectura del Software:

El propósito es describir la arquitectura del software completa sin detalles internos. Incluye las dependencias intermódulos. Es usado como ayuda en el trazado de los requerimientos: requerimientos del software pueden ser alocados y trazados desde los módulos individuales mostrados en este diagrama.

Cartas de Estructura:

Describen los flujos de control y de datos de un conjunto de subprogramas que son ejecutados secuencialmente, y muestra la organización jerárquica de llamadas de estos subprogramas.

RTM - Matriz de Traceability de Requerimientos:

Representa la alocaión de los requerimientos del software en los módulos identificados. Los requerimientos del software son listados sobre un eje y los módulos sobre el otro.

PDL - Lenguaje de Diseño de Programas:

Es usado para describir cada componente del diseño del software que resultará en código actual, como subprogramas o paquetes.

Diseño Detallado

Cartas de Estructura:

Describen los flujos de control y de datos de un conjunto de subprogramas que son ejecutados secuencialmente, y muestra la organización jerárquica de estos subprogramas.

SAD - Diagrama de la Arquitectura del Software:

Idem diseño top-level.

PDL - Lenguaje de Diseño de Programas:

Idem diseño top-level.

RTM - Matriz de Traceability de Requerimientos:

La diferencia con el diseño top-level es que los requerimientos deben ser alocados y trazados desde subprogramas individuales dentro de un módulo en adición al módulo en su totalidad.

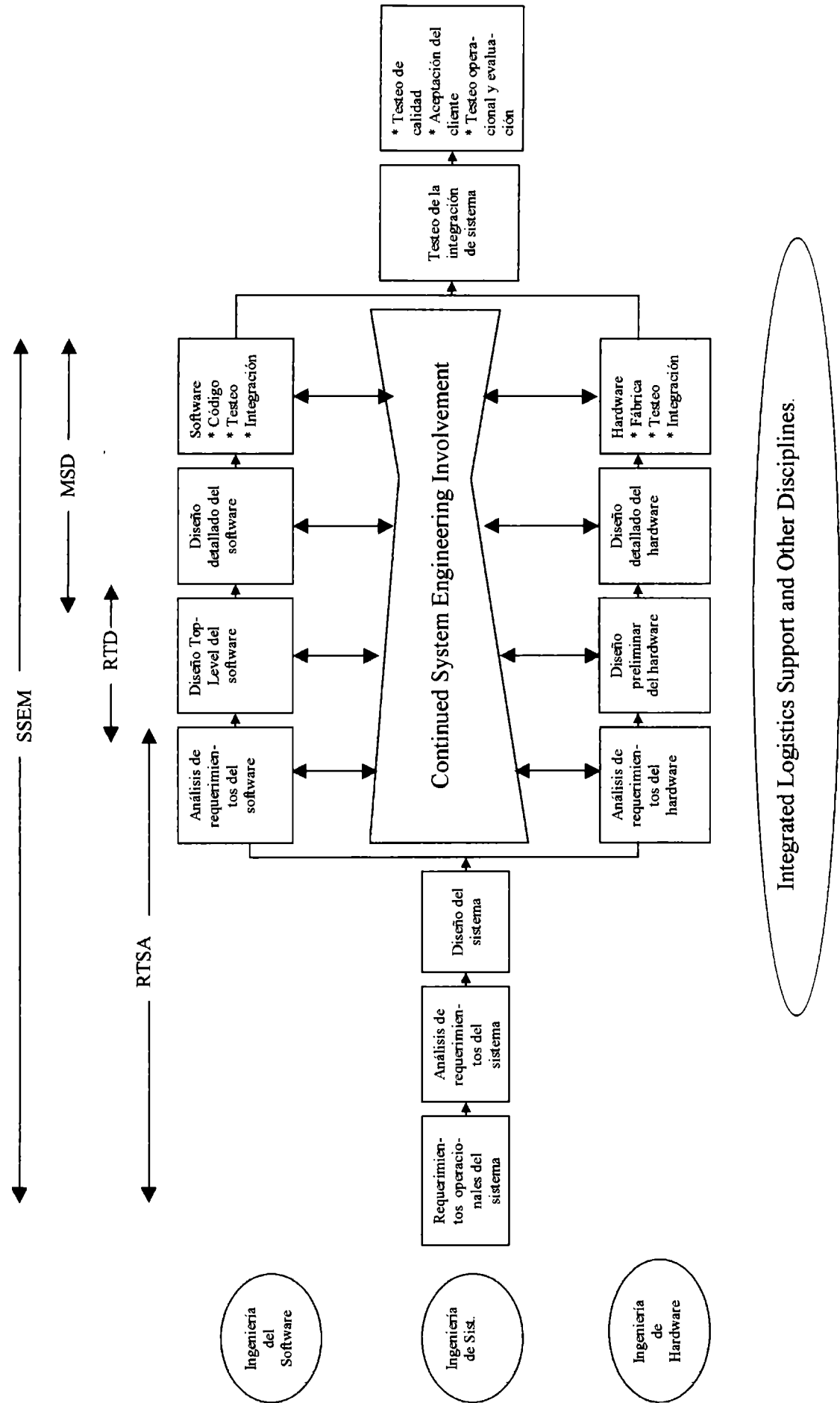


Figura 1.1

Capítulo 4

Redes de Petri

REDES DE PETRI

1. Introducción

Un modelo es una representación de las características importantes del sistema que se está estudiando. Los sistemas pueden ser muy complejos y grandes, y tener componentes que interactúan. Cada componente puede ser un sistema en sí mismo, y se puede describir independientemente con excepción de las interacciones con otros componentes. Se observa entonces que como los componentes de un sistema interactúan se hace necesaria una sincronización. Las redes de Petri fueron diseñadas especialmente para modelar este tipo de sistemas. Petri formuló la base de una teoría de comunicación entre componentes asincrónicos.

Las redes de Petri constituyen un formalismo gráfico para la especificación operacional.

2. Redes de Petri

2.1. Definiciones Básicas

La estructura de una red de Petri C es una 4-tupla, $C = (S, T, I, O)$, donde:

- $S = \{s_1, s_2, \dots, s_n\}$ es un conjunto finito de sitios, $n \geq 0$;
- $T = \{t_1, t_2, \dots, t_m\}$ es un conjunto finito de transiciones, $m \geq 0$; el conjunto de sitios y el conjunto de transiciones son disjuntos, $S \cap T = \emptyset$;
- $I : T \rightarrow S^\infty$, es la función de entrada, mapeo de transiciones a bolsas de sitios;
- $O : T \rightarrow S^\infty$, es la función de salida, mapeo de transiciones a bolsas de sitios;

Un sitio s_i es un sitio de entrada a una transición t_j si $s_i \in I(t_j)$; s_i es un sitio de salida si $s_i \in O(t_j)$. Las entradas y salidas de una transición son bolsas de sitios. Una bolsa es una generalización de conjuntos, la cual permite múltiples ocurrencias de un elemento. La multiplicidad de un sitio de entrada s_i para una transición t_j es el número de ocurrencias del sitio en la bolsa de entrada de t_j , $\#(s_i, I(t_j))$. Análogamente, para un sitio de salida la multiplicidad es $\#(s_i, O(t_j))$.

Una transición t_j es una transición de entrada a un sitio s_i si s_i es un sitio de salida de t_j . Una transición t_j es una transición de salida del sitio s_i si s_i es un sitio de entrada a t_j .

2.2. Grafo en la Red de Petri

La mayoría de los trabajos sobre redes de Petri se basan en la definición formal dada anteriormente. Sin embargo una representación gráfica de la estructura de la red es mejor para ilustrar los conceptos de la teoría de redes de Petri.

La estructura de una red de Petri se puede representar como un multigrafo dirigido bipartito. Una red de Petri es un multigrafo porque permite múltiples arcos de un nodo del grafo a otro. Además, como los arcos son dirigidos es un multigrafo dirigido. Debido a que el conjunto de nodos del grafo se puede particionar en dos subconjuntos (sitios y transiciones), tal que, cada arco se dirige de un elemento de un subconjunto a un elemento del otro, es un multigrafo dirigido bipartito.

Un arco dirigido de un sitio s_i a una transición t_j define un sitio de entrada a la transición. La definición de múltiples entradas a una transición se indica con arcos múltiples desde los sitios de

entrada hasta la transición. Un sitio de salida se indica con un arco desde la transición hacia el sitio. Nuevamente, múltiples salidas se representan con múltiples arcos. Los nodos de tipo sitio se representan con círculos, mientras que los nodos de tipo transición se representan con barras.

Un grafo de red de Petri G , es la 2-tupla $G = (V, A)$ donde:

- $V = \{v_1, v_2, \dots, v_s\}$ es un conjunto de vértices;
- $A = \{a_1, a_2, \dots, a_r\}$ es una bolsa de arcos dirigidos, $a_i = (v_j, v_k)$, tal que $v_j, v_k \in V$.

El conjunto V se puede particionar en dos conjuntos disjuntos S y T , tal que $V = S \cup T$, $S \cap T = \emptyset$.

Una marca μ es una asignación de “tokens” a los sitios de la red de Petri. Un token es un concepto primitivo de las redes de Petri, como los sitios y las transiciones. Los tokens se asignan y se puede pensar que residen en los sitios de la red.

Una red de Petri marcada M , es $M = (C, \mu)$, donde C es una red de Petri y μ es una marca. El número y la posición de tokens puede cambiar durante la ejecución de la red. Los tokens se usan para definir la ejecución de la red.

2.3. Reglas de ejecución de una red de Petri marcada

La ejecución de la red de Petri está controlada por el número y distribución de tokens en la red. Una transición está “habilitada” para dispararse si cada uno de sus sitios de entrada, tiene al menos tantos tokens como arcos del sitio a la transición. Dada una marca μ , cuando una transición se dispara se remueven los tokens de los sitios de entrada y se crean nuevos tokens que se distribuyen en los sitios de salida, resultando una nueva marca μ' . Si no hay suficientes tokens en algún sitio de entrada, la transición no está habilitada y no podrá dispararse.

3. Extensiones de las redes de Petri para modelar sistemas de Tiempo Real

Las características de los sistemas de Tiempo Real refuerzan la importancia de las herramientas de especificación y validación. Para poder utilizar las redes de Petri en la modelización de estos sistemas se necesitan incorporar restricciones de tiempo a las transiciones y nuevas clases de arcos, para poder simular condiciones de funcionamiento que con las redes de Petri clásicas no es posible.

3.1. Extensiones de tiempo

Con estas extensiones las redes de Petri serán aptas para modelar distintos fenómenos que involucran restricciones de tiempo. Entre estos fenómenos se tienen: tiempos entre llegadas de eventos externos, tiempos de reacción, duración de actividades, tiempos de timeout, acciones periódicas, etc.

Cada transición tiene asociada una restricción de tiempo. Cada restricción r se compone de una distribución probabilística d con sus parámetros, $r = (d, \text{parámetros})$. d es la distribución de la duración del tiempo que transcurre desde que la transición se habilita hasta que se ejecuta.

La restricción de una transición indica que una vez que esta se habilita, el instante en que se ejecutará se determina en función de la distribución asociada. Si se restringe lo anterior, y se utilizan sólo distribuciones uniformes, se tiene que: cada transición t tiene asignados dos valores (a, b) , parámetros de la distribución uniforme, que componen la restricción de t , tal que

$a \in [0, \infty)$, $b \in [0, \infty)$ y $b \geq a$. La transición deberá dispararse en el tiempo $T_{\text{ejecución}} = I + \delta$; tal que $I \in [a, b]$, generado con probabilidad uniforme, siendo δ el instante en que la transición se habilita.

Notar que las redes de Petri clásicas corresponderían al modelo restringido mencionado anteriormente, cuando $a = 0$ y $b = \infty$. Por lo tanto, cuando una transición no tiene restricción se toma por defecto una distribución uniforme con parámetros $0, \infty$.

La ejecución de una transición no lleva tiempo (tiempo 0), es decir la ejecución es instantánea. Para modelar una acción con cierta duración > 0 y una restricción $r = (d, \text{parámetros})$, se la desglosa en dos transiciones y un sitio intermedio. La primera transición “comenzar acción” tendrá como restricción a . El sitio será el que indique “ejecutando acción”. La última transición “fin de acción” tendrá como restricción a (uniforme, duración, duración), en el caso que la duración sea fija. La duración de una acción puede responder a una cierta distribución d' con parámetros $\text{parámetros}'$. Entonces, la transición “fin de acción” tendrá una restricción $r' = (d', \text{parámetros}')$. Si la acción es atómica, del sitio “ejecutando acción” no debiera salir ningún arco, a excepción del dirigido a la transición “fin de acción”.

3.2. Extensiones con respecto a los arcos

Los arcos de una red de Petri clásica corresponden al modelo de arcos eliminadores, es decir que su funcionamiento en la ejecución de la red consiste en eliminar un token de su sitio de origen.

Una primera extensión en las redes clásicas con respecto a los arcos es admitir arcos eliminadores con multiplicidad n , a fin de eliminar n tokens del sitio origen o depositar n tokens en el sitio destino. Esta extensión no representa un cambio conceptual sino una simplificación gráfica de la red.

Se incorporan dos clases de arcos particularmente útiles en el análisis de sistemas reales: arcos examinadores y arcos inhibidores.

El arco examinador (-----►) también simplifica el gráfico. Este se diferencia del arco eliminador en que no remueve los tokens del sitio de origen. Si varias transiciones examinan el mismo estado, esto no implica que estén en conflicto. Es decir, que esta extensión permite modelar el testeo de condiciones y no conflictúa con un posible arco eliminador vinculando los mismos sitios. El arco examinador también puede tener multiplicidad $n \geq 1$, con el fin de testear la existencia de n tokens en el sitio de origen para habilitar a la transición correspondiente. Notar que un arco examinador a la salida de una transición carece de sentido.

Una carencia de las redes de Petri es la inhabilidad de testear por exactamente una marca cero en algún sitio ilimitado¹. Con la extensión de arco inhibidor (-----○►) se permite el testeo por cero (ningún token en el sitio), ampliando la potencia de las redes de Petri. Al igual que el arco examinador si varias transiciones testean por cero un sitio no implican que estén en conflicto. Aquí tampoco tiene sentido un arco inhibidor a la salida de una transición.

Si un sitio s_i fuera limitado², con límite k y se quiere testear por cero, se puede modelar esto teniendo otro sitio s_j que mantenga el complemento de s_i . Es decir, que $\mu_j = k - \mu_i$. De esta forma para testear que s_i tenga cero tokens se testea que s_j tenga k tokens.

Así como el arco inhibidor detallado testea que no haya uno o más tokens en el sitio de origen, el arco inhibidor con multiplicidad n testea que no haya n o más tokens en el sitio de origen. Esto solo contribuye a la simplicidad de la gráfica, no aumenta la potencia de la red.

¹ Sitio cuya marca, durante la ejecución, puede tener cualquier valor natural.

² Sitio cuya marca, durante la ejecución, es siempre menor o igual a una constante natural.

3.3. Definición Formal de la Red de Petri Extendida

Algunas definiciones previas:

- $D = \{x / x \text{ es una distribución probabilística}\}$, conjunto de las distribuciones probabilísticas.
- $TA = \{\text{ELIMINADOR, EXAMINADOR, INHIBIDOR}\}$, conjunto de todos los posibles tipos de arcos.

La estructura de una red de Petri extendida C , es una 5-tupla, $C = (S, T, A, r, \mu)$; donde:

- $S = \{s_1, s_2, \dots, s_n\}$ es un conjunto finito de sitios, $n \geq 0$;
- $T = \{t_1, t_2, \dots, t_m\}$ es un conjunto finito de transiciones, $m \geq 0$; el conjunto de sitios y el conjunto de transiciones son disjuntos, $S \cap T = \emptyset$; $V = S \cup T$ es el conjunto de vértices.
- $A = \{a_1, a_2, \dots, a_k\}$ es un conjunto finito de arcos dirigidos, $k \geq 0$.
- $a_i = (v_0, v_d, m, \text{tipo})$, arcos dirigidos de v_0 a v_d , tal que $v_0, v_d \in V$, $m \in \mathbb{N}$ (multiplicidad del arco), $\text{tipo} \in K$, $((v_0 \in T \text{ y } v_d \in S) \text{ o } (v_0 \in S \text{ y } v_d \in T))$ y, si $(v_0 \in T \text{ y } v_d \in S)$ entonces $\text{tipo} = \text{ELIMINADOR}$.
- $r = (r_1, r_2, \dots, r_m)$ es el vector de restricciones de la red, r_i es la restricción de t_i .
- $r_i = (d_i, \text{parámetros}_i)$, tal que $d_i \in D$ y parámetros_i son los parámetros de d_i .
- $\mu = (\mu_1, \mu_2, \dots, \mu_n)$, es el marcaje inicial de la red, $\mu_i \in \mathbb{N} \cup \{0\}$, μ_i es la marca de s_i .

4. Ejecución de las Redes de Petri Extendidas

Es de suma importancia que luego de realizar una especificación del sistema se realice una validación o verificación del modelo desarrollado.

Validar significa probar rigurosamente que un sistema dado cumple con las funciones indicadas en la especificación. La validación rigurosa y completa de un sistema no siempre es posible, especialmente cuando se trata de sistemas de tiempo real complejos, pero toda aproximación a dicha validación constituye una valiosa contribución a la confiabilidad.

La verificación es en algunos casos el nivel de validación posible: dado un sistema de interés y conjunto de estados y eventos posibles para el sistema, se estudia sistemáticamente su respuesta verificando el cumplimiento de lo indicado en la especificación. Lógicamente, la generalidad de los resultados es menor que haciendo una validación, pero la factibilidad de la verificación es habitualmente mayor que la de validar formalmente.

Mediante la ejecución de la red de Petri extendida es posible lograr una simulación del comportamiento del sistema modelado y, de ésta forma, verificar el modelo planteado. Otra de las ventajas de poder simular la ejecución del sistema, es que se pueden experimentar variaciones en el modelo y observar los resultados, cosa que generalmente es muy costosa o imposible de realizar sobre el sistema real. La simulación permite, además, hacer predicciones acerca del comportamiento del sistema, con las que se pueden decidir cambios en el modelo.

4.1. Reglas de ejecución

La ejecución de una red de Petri extendida está controlada no solo por el número y distribución de tokens en los sitios, sino también por el paso del tiempo. Es así que una transición que está habilitada para dispararse, solo se disparará en un instante de tiempo determinado por su

restricción. De esta manera la red se ejecutará de la forma mas próxima al comportamiento del sistema modelado.

Una transición está habilitada cuando se cumplen todas sus condiciones. Las condiciones se determinan teniendo en cuenta los arcos de entrada a la transición. Para cada arco de entrada, si el arco es de tipo eliminador o examinador, la cantidad de tokens en el sitio de origen debe ser mayor o igual que la multiplicidad del arco. Si el arco es de tipo inhibidor, la cantidad de tokens en el sitio de origen debe ser menor que la multiplicidad del arco.

Una transición puede estar habilitada para dispararse n veces. Esto es así cuando la transición está habilitada y por cada arco eliminador entrante la cantidad de tokens en el sitio de origen es n por la multiplicidad del arco.

En el sistema puede haber acciones que se ejecuten sin condiciones, por lo tanto en la red habrá transiciones sin arcos entrantes. Estas transiciones están siempre habilitadas una vez. Es decir, inicialmente estan una vez habilitadas, y cada vez que se ejecutan se vuelven a habilitar.

En las redes de Petri clásicas, la elección de la transición a dispararse, de entre todas las habilitadas, es no determinística. En las redes de Petri extendidas, en esta elección influyen las restricciones de tiempo de las transiciones habilitadas, eligiéndose la que tenga el instante de ejecución mas temprano. Así, se elimina en parte, el no determinismo propio de las redes de Petri.

La ejecución de una transición no insume tiempo, es decir es instantánea. Cuando, dada una marca μ , una transición t_i se dispara, resulta una nueva marca μ' . Esta nueva marca se genera teniendo en cuenta los arcos de entrada y salida de t_i . Por una parte, por cada arco de entrada a t_i que sea de tipo eliminador se resta a la marca del sitio de origen la multiplicidad del arco. Por otra parte, por cada arco de salida de t_i se suma la multiplicidad del arco al sitio destino. Notar que los sitios de origen de los arcos de tipo examinadores e inhibidores entrantes a t_i no modifican su marca.

4.2. Trama o secuencia de la ejecución

Dada una red de Petri extendida $C = (S, T, A, \tau, \mu_0)$, μ_0 es la marca inicial. Se ejecuta esta red por el sucesivo disparo de sus transiciones. Al disparar una transición habilitada t_j con la marca inicial, se produce una nueva marca μ_1 , por esto $\mu_1 = f(\mu_0, t_j)$. Con esta nueva marca, se puede disparar cualquier otra transición habilitada, supongamos que t_k es la que tiene el instante de ejecución mas temprano, entonces resulta una nueva marca $\mu_2 = f(\mu_1, t_k)$. Esto se puede continuar mientras haya al menos una transición habilitada. Si se alcanza una marca μ_t para la cual no hay una transición habilitada, la ejecución se bloque y termina, es decir, que $f(\mu_t, t_x)$ no está definida para ninguna transición t_x .

$f: N^n \times T \rightarrow N^n$ se denomina función de próximo estado. Observar que cada estado del sistema queda definido por el marcaje de la red.

Cada vez que se dispara una transición y la red cambia de estado (ya que cambia el marcaje), se ejecuta un paso en la ejecución de la red. Cada paso está determinado por la marca que permitió la ejecución de la transición, por la transición que se ejecutó y por el instante en que se ejecutó. La cadena de pasos ordenados según el instante de ejecución se llama "trama o secuencia de una ejecución" :

TRAMA = $(\mu_0, t_0, \text{tejecución}_0), (\mu_1, t_1, \text{tejecución}_1), (\mu_2, t_2, \text{tejecución}_2), \dots, (\mu_h, t_h, \text{tejecución}_h)$.

La trama podría llegar a ser infinita si la ejecución no termina. Normalmente las simulaciones se realizan por un periodo de tiempo determinado, por lo que las tramas de dichas ejecuciones quedan interrumpidas. Las tramas generadas por una misma red de Petri extendida pueden ser distintas.

Capítulo 5

**Sistemas de Información
Geográfica**

SISTEMAS DE INFORMACIÓN GEOGRÁFICA – G.I.S.

1. ¿Qué es un Sistema de Información Geográfica?

- Un Sistema de Información Geográfica es una herramienta basada en computadoras para mapear y analizar cosas que existen y eventos que suceden sobre la Tierra. La tecnología GIS integra operaciones de Bases de Datos comunes como consultas y análisis estadísticos con la visualización y los beneficios de análisis geográficos ofrecidos por los mapas. Estas habilidades distinguen a los GIS de otros sistemas de información y hacen esto valioso para un amplio rango de empresas públicas y privadas para explicar eventos, pronosticar resultados, y planear estrategias.
- Un Sistema de Información Geográfica está diseñado para trabajar con datos referenciados por coordenadas geográficas o espaciales. En otras palabras, un GIS es tanto un Sistema de Base de Datos con capacidades específicas para datos referenciados espacialmente, como también un conjunto de operaciones para trabajar con los datos. (Star and Estes, 1990)
- Un Sistema para capturar, almacenar, chequear, integrar, manipular, analizar y mostrar datos que son referenciados espacialmente en la Tierra. (Chorley, 1987)
- Sistemas automatizados para la captura, almacenamiento, recuperación, análisis y mostrado de datos espaciales. (Clarke, 1990)
- Un Sistema de hardware, software y procedimientos diseñados para soportar la captura, manejo, manipulación, análisis, modelado y visualización de datos referenciados espacialmente para resolver problemas complejos de manejo y planeamiento. (NCGIA lecture by David Cowen, 1989)
- Los sistemas de Información Geográfica son sistemas basados en computadoras, usados para almacenar y manipular información geográfica. Es el resultado del deseo y la necesidad de ejercer todo tipo de controles sobre procesos ambientales, económicos y sociales.
- En sus principios, la tecnología G.I.S. se limitaba a superponer y combinar distintos tipos de datos en un mapa. Hoy en día, es mucho más que un sistema CAD, dado que asocia atributos geográficos y no geográficos de recursos cartográficos. Los mapas resultantes muestran datos geográficos, ambientales o demográficos separadamente o en conjunto.
- Caso especial de los sistemas de información donde la base de datos consiste en las obtenciones de las características, actividades o eventos distribuidos espacialmente, los cuales están definidos en el espacio como punto, línea o área. Un G.I.S manipula los datos sobre estos puntos, líneas y áreas para la recuperación de atributos y análisis.
- Un conjunto automático de funciones que provee a los profesionales especializados el almacenamiento, recuperación, manipulación y reproducción gráfica de los datos localizados geográficamente.

- Sistemas o herramientas asistidas por computadora para captura, almacenamiento, transformación análisis y reproducción gráfica de datos espaciales.

2. Subsistemas de un G.I.S.

- **Subsistema de Procesamiento de datos:**

- Adquisición de datos: Mapas, imágenes, etc.
- Entrada de datos: Ingreso de información a la base de datos.
- Almacenamiento de datos: Frecuencia de uso, políticas de uso, actualización y seguridad.

- **Subsistema de Análisis de datos:**

- Recuperación y Análisis: Para dar respuesta a preguntas de todo tipo (espaciales o no) teniendo en cuenta los datos almacenados.
- Salida de información: Para mostrar resultados (mapas, tablas, etc.) o para alimentar algún otro sistema digital.

- **Subsistema de uso de la información:**

- Incluye a los usuarios que pueden ser investigadores, planificadores, gerentes, etc.
- Debe haber una interacción entre el personal de desarrollo del G.I.S. y los usuarios, a fin de planificar la funcionalidad y las estructuras de datos.

- **Subsistema Administración:**

- Estructura Organizacional: el sector de G.I.S. muchas veces está ubicado en forma separada del resto de las partes de la Organización.
- Equipo Humano: incluye al gerente de sistemas, al administrador de la base de datos, el analista de sistemas y al operador de digitalización.
- Procedimientos de administración: Incluye las pautas a seguir para una eficiente interacción entre el grupo del G.I.S. y los usuarios. La eficiencia en la interacción contribuye a la eficiencia del sistema.

3. Principales Formas de Uso y Aplicaciones de un G.I.S.

- **Formas de uso:**

- En la administración de servicios, recursos, propiedades y proyectos, el G.I.S se usa para la toma de decisiones.
- En estudios científicos, es un gran aporte para el análisis estadístico.
- En integración de áreas de conocimiento colabora en la recolección y visualización de datos espaciales.

- **Áreas de aplicación:**

- Aplicaciones comerciales: por la facilidad de retratar los datos geodemográficos y de poder analizarlos brinda la posibilidad de interacción comercial.
- Administración política: determinación de zonas electorales para la manipulación geodemográfica y decisiones políticas.
- Administración de infraestructuras: contribuye a la planificación de redes cloacales, de agua, servicios eléctricos, televisión por cable, etc.
- Publicación de bases de datos y mapas: se utiliza para este fin en Departamentos estatales, Municipios, empresas privadas, etc.
- Exploración de zonas petroleras, de gas y de minerales.
- Seguridad pública: combatiendo crímenes, robos, incendios y epidemias.
- Administración de información en Tiempo Real: brinda información de este tipo para ser usada en las diversas áreas del conocimiento.
- Administración de Recursos Naturales: estudios y proyecciones sobre el uso de agua, contaminación del aire, etc.
- Transporte y apoyo logístico: aconsejando los lugares por donde es conveniente construir una nueva ruta, playa de estacionamiento, hospital, escuela, etc. Utiliza para esto los datos geodemográficos.

Capítulo 6

Agemap 3.5

AGEMAP 3.5

1. Introducción

Agemap es un G.I.S. (Geographic Information System). Un sistema de información geográfica que utiliza información geográfica digital y con él se puede manipular diferentes mapas, enlazar datos contenidos en bases de datos Xbase para visualizarlos y procesarlos geográficamente.

2. Mapas

El programa maneja los mapas como una biblioteca por lo que se pueden agregar más mapas a los ya existentes sin tener un límite máximo, más que el propio espacio en disco. De cada mapa se puede abrir hasta un máximo de dos ventanas que se trabajan de forma independiente para poder tener a la vista dos partes distintas de la ciudad a diferentes escalas.

Desde la ventana del mapa se puede consultar un elemento de éste especificando primero que es lo que se va a consultar (por ejemplo: nombre de calles, numeración, objetos de la base de datos, zonas, etc.).

Agemap tiene incluida la numeración de las calles en cada esquina; de esta forma es como se ubican las direcciones en la forma calle- número.

3. Consulta de elementos de una Base de Datos

Agemap permite el enlace de bases de datos al mapa de forma de poder realizar consultas geográficas de esos datos mediante la representación de los registros de esa base de datos con iconos.

4. Distancias

Agemap permite realizar mediciones sobre el mapa.

5. Barrios

Cada mapa puede tener incorporada la información de los barrios, dependiendo de la información disponible al momento de digitalizarlo. La inclusión de esta información y la posibilidad de su modificación hace que el sistema permita una adaptabilidad mayor en el momento de ubicar direcciones.

Cada barrio está digitalizado como una línea poligonal cerrada y es posible crear nuevos barrios o divisiones distintas o modificar las ya existentes y luego serán utilizadas en el momento del enlace de la base de datos con el mapa.

6. Bases de Datos

Agemap cuenta con un manejador de base de datos que provee las mismas funciones, comandos y opciones que cualquier programa Xbase. Esto posibilita la creación de bases con formato DBF, las cuales se pueden modificar, ordenar, buscar cierto dato, crear índices de búsqueda, entre otras.

Al igual que en los programas Xbase, éste sistema funciona sobre la base de áreas de trabajo en las cuales se manipulan las diferentes bases de datos, es decir que los comandos y funciones que se utilicen serán aplicadas al área de trabajo que se encuentre seleccionada en ese momento.

Cada base de datos podrá ser utilizada en un área de trabajo, es decir que se puede mantener abiertas diferentes bases de datos a la vez pero solo una de ellas podrá estar activa para poder efectuar modificaciones u otras acciones sobre ella.

La opción Registros/Preferencias Geográficas permite al usuario determinar la forma de visualización de los objetos de la base de datos y el tipo de información que contienen los campos de ésta. Este diálogo también permite conectar una base de datos con un mapa, de forma de representar en forma gráfica, la información contenida en dicha base.

Para que Agemap pueda extraer o almacenar datos de la base es necesario especificar que campo se destina a que fin. Estos campos tendrán asignados los llamados “Geotipos”, que informan a Agemap de la naturaleza del contenido del mismo.

Luego de hecho esto, Agemap generará el enlace interno de la base de datos. Concluido esto la base de datos está lista para generar las coordenadas geográficas de las direcciones de los registros luego de lo cual se podrán visualizar los objetos sobre el mapa. Para esto se emplea el proceso Geocodizador.

7. Geotipos

Los Geotipos son tipos especiales que se asignan a los campos de la base de datos y le indican a Agemap cual es la información contenida en el campo o la función de él; es decir, que al igual que cuando se le indica a un programa Xbase que un campo va a tener determinadas características (Carácter o Numérico o tener Decimales), en Agemap, se debe incluir una nueva característica al campo que es el Geotipo que ese campo va a tener.

Los tres Geotipos mínimos necesarios que se deben asignar para que se logre el despliegue de los datos en el mapa son los de Coordenada X, Coordenada Y y Dirección. De ésta forma el programa tiene la información básica necesaria para trabajar; el campo de donde tomar la dirección y los campos donde colocar las coordenadas X, Y para uso propio del sistema.

En la siguiente tabla aparecen todos los Geotipos con el tipo de campo de la base al cual se debe relacionar, el ancho mínimo de ese campo y de que tipo es.

Existen tres tipos:

Sistema: El campo de la base que tiene asignado este Geotipo es para uso exclusivo de Agemap, en él se colocan datos que son usados por el sistema (por ejemplo, las coordenadas).

Entrada: Los campos con Geotipos de entrada se refiere a que son extraídos datos desde esos campos, la naturaleza de los datos depende del Geotipo (por ejemplo, la dirección necesaria para ubicar al registro en el mapa se toma del campo con Geotipo Dirección)

Retorno: En estos campos Agemap deja información dependiendo del Geotipo asignado (por ejemplo, luego de Geocodizar Agemap deja el código postal de la dirección del registro en el campo asignado al Geotipo Código Postal)

Es necesario crear dos campos numéricos de al menos cinco posiciones de largo y 0 decimales que el sistema utilizará para asignar coordenadas X, Y de despliegue para cada registro.

Nombre de Geotipo	Tipo del campo	Largo mínimo	Tipo
0 – Sin Relación	No aplicable	No aplicable	No aplicable
1 - Coordenada X	Numérico	5	Sistema
2 - Coordenada Y	Numérico	5	Sistema
3 - Calle Principal	Numérico	5	Retorno
4 - Primera Esquina	Numérico	5	Retorno
5 - Segunda Esquina	Numérico	5	Retorno
6 - Dirección	Carácter	Cualquiera	Entrada
7 - Esquina	Carácter	Cualquiera	Entrada
8 - Barrio	Carácter	Cualquiera	Entrada
9 - Numero de Puerta	Numérico o carácter	Cualquiera	Entrada
10 - Código Postal	Carácter	6	Retorno
11 - Resultado	Numérico	2	Retorno
12 - Campo de despliegue	Numérico o carácter		Entrada
13 - Icono	Numérico o carácter	Cualquiera	Entrada
14 - Color Icono	Numérico	1	Entrada
15 - Tamaño de Icono	Numérico	1	Entrada
16 - Código de Zona	Carácter	4	Retorno
17 - Código de Trayecto	Carácter	4	Retorno
18 - Orden de Trayecto	Numérico	3	Retorno
19 - Radio de Influencia	Numérico	1	Entrada
20 - Tiempo de Espera	Numérico	1	Entrada
21 - Código de Manzana	Carácter	6	Entrada
22 - Nombre de Barrio	Carácter	6	Retorno

A continuación se detalla la forma de utilización de estos Geotipos:

a) Sin Relación: El campo no tiene ninguna relación con el sistema de modo que los datos contenidos en ellos no serán tocados por él, por defecto todos los campos de la base de datos tienen este Geotipo.

b) Coordenada X: Este campo es para uso exclusivo del sistema; en él se colocan las coordenadas de despliegue de ese registro. Estas coordenadas son propias de cada mapa y aquellos registros que contengan coordenadas 0,0 significa que no están ubicados en el mapa.

c) Coordenada Y: Idéntico al Geotipo de Coordenada X

d) Calle Principal: Agemap maneja internamente un código para cada calle del mapa. Si éste Geotipo está asignado a un campo de la base, el geocodificador dejará el código de la calle principal encontrada en ese campo y cuando se utiliza el diálogo de consulta especializada para posicionar un registro también es utilizado éste geotipo para dejar la información del código de la calle principal. En la programación de Agemap puede utilizarse ésta información para acelerar algunos procesos ya que existen algunos comandos y funciones que se basan en este código.

e) Primera esquina: Idéntico al Geotipo de Calle Principal. Se toma como primera esquina aquella que se encuentra más cerca de la dirección especificada.

f) Segunda Esquina: Idéntico al Geotipo de Primera Esquina

g) Dirección: De este campo Agemap toma la información de la dirección que utiliza el geocodificador. La Geocodificación es el proceso que genera las coordenadas a partir de la información de éste campo, para ubicar cada uno de los registros. La forma más usual de ingresar la información es colocar toda la dirección en un solo campo (calle, número de puerta, esquina, barrio, etc.), el geocodificador posee una pseudo-inteligencia que permite encontrar las diferentes partes de la dirección dentro del campo. En caso de tener la información de la dirección en más de un campo referirse a los Geotipos de Esquina, Número de puerta, Barrio. Si el barrio no tiene especificado delimitador dentro de la dirección, se considerará como barrio todas las palabras que se encuentren desde el final del campo hasta el primer número.

h) Esquina: Este Geotipo es necesario en caso de tener la dirección en más de un campo ya que con él se indica donde está el nombre de la calle secundaria o esquina. Cuando la base de datos tiene un campo para la calle principal y otro para la esquina, se debe asignar el Geotipo de dirección al campo que contiene la calle principal y el de Esquina al que tiene la calle secundaria

i) Barrio: Es usado para indicar que el campo contiene un filtro geográfico (por ejemplo, barrio, partido, estados, colonias, municipios, etc. Esto es necesario para poder diferenciar calles con el mismo nombre pero que pertenecen a distintas localidades.

j) Número de puerta: Se asigna a un campo este Geotipo cuando contiene el número de puerta de la dirección.

k) Código Postal: Este Geotipo indica dónde se dejará el código postal luego de geocodificado el registro. Este Geotipo es del tipo de Retorno porque algunos procesos dejan información en el campo que tenga esta relación. También es posible que funcione como Geotipo de Entrada ya que, en algunos casos que dependen del mapa, el código postal es utilizado para Geocodizar los registros.

l) Resultado: En este campo se deja un número que corresponde a una clave. Permite saber cual fue el resultado de la geocodificación de ese registro y el tipo de falla en caso de no haber podido geocodizar correctamente ese registro.

m) Campo de Despliegue: Cuando éste Geotipo está asignado se despliega el contenido del campo debajo del icono en el mapa. De ésta forma se puede identificar al objeto de la base mediante el despliegue de un código o nombre corto. Se recomienda que el contenido del campo no sea extenso porque es posible que se oculte una buena parte del plano en las escalas más altas.

n) Icono: Este Geotipo permite que se desplieguen distintos iconos para cada registro. Cuando está asignado el Geotipo, el sistema busca en ese campo un identificador de icono. Si el campo tiene una cadena de caracteres el sistema desplegará ese registro con el icono que tiene ese nombre. Si el campo contiene un número se despliega el icono que tiene ese número. Los iconos se dibujan y modifican desde el programa auxiliar Generador de Iconos que se incluye con Agemap y en él es posible indicar un nombre además del número que lo identifica y con lo cual será referenciado con éste geotipo. En caso de no poder encontrar icono con ese identificador se usará el icono por defecto que se especificó en el dialogo de Preferencias geográficas. De esta forma se pueden visualizar los clientes de distintos vendedores, las empresas que venden distintos productos o el tipo de contrato con esas personas mediante diferentes iconos por más que pertenezcan a una sola base de datos.

o) Color de icono: Este Geotipo tiene relación con el de Icono y se puede variar el color del icono dependiendo del valor que se encuentre en el campo asignado del registro que se está considerando; éste valor puede variar en la escala de colores del 0 al 15 y mediante la variación de estos colores es posible visualizar los diferentes clientes que cumplan con determinadas características que el usuario desee resaltar y que luego podrá visualizar rápidamente sobre el mapa. Un caso típico de la utilización de éste Geotipo es cuando se tiene

en una base de datos diferentes tipos de comercios y se desea representar alguna diferencia que exista entre ellos, por ejemplo la venta que realiza cada uno de ellos.

p) Tamaño de icono: Este Geotipo es similar en su concepto al de Color de Icono y los valores o las escala de colores permitidos es de 0 a 100. Puede variar el tamaño de cada icono dependiendo del valor que se encuentre en el campo asignado. Esto posibilita visualizar rápidamente datos como importancia del cliente, veces que se solicito el servicio, volumen de ventas, etc., mediante el tamaño proporcional del icono. Se debe tener cuidado porque el tamaño máximo es de 100 unidades en caso de sobrepasar dicho numero se utilizará el tamaño por defecto que es 10. La relación de tamaño entre los iconos se representará de manera que 10 sea el tamaño normal de éste, de 10 a 100 se efectuarán aumentos de tamaño de manera tal que la relación de 100 sea 10 veces más grande que el tamaño normal; del mismo modo cuando se representa con valores menores que 10 se disminuirá el tamaño del icono; es decir que si tomamos un valor de 1 significará una disminución de tamaño de 10 veces más pequeño.

q) Código de Zona: Este Geotipo le indica al sistema en donde debe colocar el código de zona a la cual pertenece el registro. Una vez creada una zona y asignado este Geotipo existe un proceso por el cual el sistema coloca en ese campo el código de la zona a la cual pertenece el objeto; por ejemplo, supongamos que existen zonas de venta atendidas por vendedores y se tiene una base de datos con los lugares a ser visitados, se podrán asignar los códigos de zonas que corresponden a cada uno de los registros de manera que luego es posible listarlos ordenados por zona de venta.

r) Código de Trayecto: Es similar al de Código de Zona y coloca el código del trayecto a los registros que se encuentran sobre algún trayecto determinado.

s) Orden de Trayecto: Asignado el Geotipo de Código de Trayecto en el campo que contenga éste Geotipo se colocará el número de orden que le corresponde. Este número es el orden de aparición de como se encuentra sobre el trayecto, debido a que cuando se crean los trayectos se indica el punto de partida y el punto de llegada de éste y por eso es posible ordenarlo ya que el primer punto es el que se encuentra más cerca del comienzo del trayecto y el último el que se encuentra más cerca del final. De ésta forma se ordena a las personas o empresas que se encuentran sobre un trayecto determinado en la forma en que se irán encontrando a medida que se recorra ese trayecto.

t) Radio de Influencia: Los datos que se encuentran en el campo asignado con éste Geotipo son los que se utilizarán para generar los mapas de Densidad. Estos mapas colorean el área alrededor de cada objeto de la base de datos con una aureola tan intensa y grande dependiendo de la relación con los números de ese campo, ya que en el momento de representación de los mapas de Densidad se indica como es la relación de los valores contenidos en los registros representados en metros. Esto permite visualizar los lugares en que son de buena venta y los que no, áreas donde la cantidad de pedidos son mayores y no existe distribuidor en ese lugar, etc. Está es una herramienta dirigida al área de Marketing y búsqueda de Mercados.

u) Tiempo de Espera: El campo asignado con éste Geotipo puede contener el tiempo (en minutos) que se calcula se estará en ese lugar. De esta forma cuando se consulta sobre el trayecto el sistema informa la distancia recorrida hasta el punto consultado desde el principio y el tiempo estimado. Los tiempos se toman en base a una tabla de velocidades de calles y a ese tiempo se le agrega el de espera de todos los puntos que se encuentran hasta ese punto consultado sobre el trayecto desde el principio del mismo. Esta información permite establecer cálculos aproximados de tiempos y distancias para los recorridos y poder optimizar las entregas, visitas, services, etc.

v) Código de Manzana: En las bases de datos de Censos de Población, de empresas de encuestas, o de Catastro la información no se encuentra en forma de direcciones (calle y número) sino por código de manzana. El sistema permite codificar las manzanas y de esa forma ubicar geográficamente los datos de esas bases. Esto permite el estudio del mercado, y

de la población existente en la ciudad para la planificación de estrategias de mercadeo, trabajos de estadística, Planeamiento e Ingeniería.

w) **Nombre de barrio:** Este Geotipo funciona de manera similar al Geotipo de Código de Zona con la diferencia que cuando se utiliza el proceso de asignación de barrios, el contenido del campo asignado queda con el nombre del barrio al cual pertenece el registro.

8. Geocodizador

Este es el proceso que genera las coordenadas geográficas para cada uno de los registros de una base de datos a partir de la información que se tenga de la dirección; de ésta forma una vez que un registro tiene coordenadas geográficas X, Y se desplegará en el mapa.

La base a Geocodizar debe tener asignados los GEOTIPOS mínimos para rastrear las direcciones y almacenar las coordenadas (Dirección, Coordenada X, Coordenada Y en caso de estar distribuida la información en varios campos son necesarios otros Geotipos)

El Geocodizador cuenta con dos modos de funcionamiento uno automático y otro interactivo.

8.1 Modo automático

Cuando se geocodiza la base de datos en modo automático el sistema procesa todos los registros que se incluyan en el ámbito y cumplan con la condición establecida (si el ámbito es ALL y no existe condición se geocodizan todos los registros). Luego se emite el reporte del resultado del proceso en la ventana de Resultados.

Ejemplo: -----
Total de registros procesados: 1542
Total de registros rechazados: 10
Otros registros rechazados: 23

En este reporte aparece el total de registros que se procesaron y además los que fueron rechazados por no poder determinar su ubicación en el plano. Primero se especifica la cantidad de registros rechazados por error en la base de datos y se refiere a la cantidad de registros que no pudo interpretar su dirección debido a que no se encontró correspondencia de ésta en la base de datos del mapa (por ejemplo, nombre mal escrito) o que existan varias posibilidades en el nombre. El segundo reporte se refiere a nombres de calle que se encontraron en la base de datos y no se pudieron posicionar por falta de datos (por ejemplo, calles que no tienen numeración o no se cruzan las calles).

8.2 Modo interactivo

Este modo deja que el usuario sea quien tome las decisiones en cuanto a definir que calle es la que corresponde en el registro que se está considerando. El sistema procesa la información y despliega una lista de calles posibles, y es el usuario quien elige la ubicación.

La forma de utilizar estos modos es de la siguiente manera. Primero se debería procesar la base de datos mediante una geocodización en modo automático; luego en caso de haber sido rechazados registros, se procede a geocodizarlos en modo interactivo pero sin geocodizar todos los registros nuevamente; solo procesando los que fueron rechazados.

9. Comandos y funciones geográficas

Geocod

Comando: Realiza la geocodización de la base de datos activa.

Sintaxis:

GEOCOD<expL1>,<expL2>,<expL3>,<expL4>,<expL5>,[<expL6>] [<expC7>]
[WHILE <expL8>] [FOR <expL9>]

Argumentos: <expL1> Geocodificación interactiva
<expL2> Check de Estricto
<expL3> Check de Todos
<expL4> Compatibiliza la base de datos si es de origen ASCII
<expL5> Check de Comienzo
<expL6> Geocodifica la base por código postal
<expC7> Alcance de la operación
<expL8> Condición del WHILE
<expL9> Condición del For

Descripción: Este comando realiza la Geocodificación de la base de datos activa y los parámetros permiten la activación de los check de configuración según los parámetros pasados.

La primera expresión identifica el check que controla el modo de Geocodificación. Si ésta expresión retorna verdadero se realiza una geocodización interactiva, caso contrario se realiza una geocodización automática. Las siguientes expresiones lógicas representan los check del diálogo del Geocodizador según su ubicación en él.

Glink

Comando: Enlaza los campos de la base de datos actual con los Geotipo correspondientes

Sintaxis: GLINK <expC1> , <expN2>

Argumentos: <expC1> Nombre del campo a enlazar
<expN2> Número del Geotipo a enlazar

Descripción: Con éste comando se enlaza el campo especificado mediante <expC1> con el Geotipo <expN2>. La lista de Geotipo así como su correspondencia con <expN2> se encuentra descrita en el capítulo de Geotipos.

Glinkm

Comando: Enlaza una base de datos con un mapa

Sintaxis: GLINKM <expC1> , <expL2> , <expN3> , <expN4> , <expN5>

Argumentos: <expC1> Identificador de mapa
<expL2> Si efectúa el enlace de la base de datos
<expN3> Icono por defecto para la base de datos
<expN4> Color del icono por defecto para la base de datos
<expN5> Tamaño del icono por defecto para la base de datos

Descripción: Enlaza la base de datos actual con el mapa especificado en <expC1>. Si <expL2> es .T. se realiza el despliegue de los registros en el mapa. En caso de ser .F. se mantiene la relación pero no se despliegan los registros de la base.

Gdistance

Función: Retorna la distancia entre dos puntos

Sintaxis: GDISTANCE(<expN1>, <expN2>, <expN3>, <expN4>)

Argumentos: <expN1> Coordenada X1 del primer punto
<expN2> Coordenada Y1 del primer punto
<expN3> Coordenada X2 del segundo punto
<expN4> Coordenada Y2 del segundo punto

Retorna: <expN> Un valor numérico que indica la distancia.

Descripción: Retorna la distancia euclidiana que existe desde el punto determinado por los parámetros <expN1> y <expN2> y el punto <expN3> y <expN4>.

10. Zonas

El dibujo de zonas le permite al usuario dividir un mapa en diferentes áreas que luego puede utilizar para efectuar potentes consultas en las bases de datos. Una zona es un dibujo sobre el mapa el cual posee un código único que la identifica, un nombre de referencia, un nivel de visualización, una textura de relleno y un color. Cuando se crea una zona, el usuario, dispone de un nuevo objeto geográfico para realizar consultas combinando los objetos de las bases de datos. De esta forma se pueden emitir reportes por zonas, contar los clientes que se encuentran en una zona específica, etc.

La definición de zonas admite adjudicar un nivel para su visualización y mediante ésta forma, se organizan las zonas ingresadas según un orden de despliegue, con lo que se puede indicar que se vean solo las zonas que pertenezcan a ciertos niveles.

11. Trayectos

Este sistema permite diseñar trayectos entre dos puntos del mapa. Los trayectos son estáticos, por esto el que crea el trayecto es el propio usuario. Los trayectos proveen información geográfica de la misma forma que las zonas, de ellos se puede obtener información sobre la distancia recorrida desde el principio hasta cualquier punto de él, el tiempo aproximado que llevaría recorrerlo, cuales registros de la base de datos se encuentran sobre él y en que orden, etc. Se puede emitir reportes de los registros de la bases de datos que están sobre determinado trayecto y de igual manera se puede emitir el orden de cómo se encontrará a dichos objetos al recorrerlo.

La creación de un trayecto consta de dos partes, la definición de sus características y el trazado propiamente dicho.

Se crea primeramente lo que llamaremos el cabezal del trayecto, en él se definen las características que lo componen como son el código, el nombre, el nivel, el color y los puntos de partida y el de llegada.

En primer lugar deberá ingresarse un código único que identificará el trayecto y puede ser de hasta cuatro caracteres alfanuméricos (letras y números); luego se deberá ingresar el nombre que tendrá el trayecto, el color con que se desplegará sobre el mapa, el nivel de visualización y si el trayecto se desplegará en el mapa o no.

Una vez ingresado estos datos se especifican los extremos del trayecto; esto significa que se debe ingresar el punto de comienzo y el punto de destino (pudiendo ser la misma dirección para ambos puntos) esto se logra mediante el ingreso de éstos puntos a través de la llamada al diálogo de Búsqueda Geográfica. Una vez ingresados éstos dos puntos estarán completos los datos que son

necesarios para el cabezal del trayecto, quedando solo pendiente el trazado de las calles que lo componen.

Una vez asignados todos los atributos al trayecto se confirma el diálogo con OK y el trayecto pasará a formar parte de la lista de trayectos; con éste método se crean todos los cabezales de las zonas, pero no su trazado en el mapa.

Luego de definido el cabezal del trayecto se trazan las calles que lo componen. Para esto se selecciona el trayecto desde el diálogo de preferencias de archivos y se presiona el botón de Dibujar.

Una vez hecho esto se ajusta automáticamente la escala del mapa para visualizar los dos puntos del trayecto marcados con una A y una B para el comienzo y el fin respectivamente y además se setea el ratón para que quede en modo de ingreso de calles. Para ingresar las calles que conforman el trayecto se debe pulsar el botón izquierdo del ratón sobre las calles por las cuales pasará; teniendo en cuenta que la calle seleccionada debe tener intersección con la anterior que conforma el trayecto; es decir que el trayecto debe tener una continuidad. Cuando se selecciona una calle esta aparece resaltada en toda su extensión mediante un trazo más grueso y de color azul; de esta forma se seleccionan las calles por las cuales pasa el trayecto.

Una vez que se termina el trayecto se debe indicar al sistema este hecho mediante la utilización de la opción del menú Trayectos/Terminar donde se indicará si se confirma las modificaciones efectuadas al trayecto.

12. Manejo de DDEML

Esta función de Agemap transforma al sistema en un servidor DDEML ya que permite ejecutar los diferentes comandos y funciones del sistema desde otra aplicación.

Todas las aplicaciones que posean la capacidad de actuar como clientes DDEML podrán utilizar ésta capacidad de Agemap y para esto, el usuario deberá consultar con los manuales de la aplicación que esté usando ya que cada aplicación en particular tiene distintos comandos y funciones para acceder a los servicios de DDEML.

El servicio con el cual conectarse es AGEMAP35 y el tópico del enlace es SYSTEM.

El acceso de DDEML se realiza estableciendo un enlace entre la aplicación y Agemap en la que la aplicación consulta al sistema Agemap mediante la utilización de cualquiera de los comandos y funciones que se dispone en el sistema; es importante tener en cuenta que Agemap debe estar ejecutándose en la terminal que efectúa las consultas.

Importante: Solo son soportadas las transacciones sincrónicas XTYP_REQUEST y XTYP_EXECUTE; recordar fijar el tiempo de espera suficiente para permitir la terminación del comando a ejecutarse.

Desde la aplicación que efectúa la consulta se debe emitir un comando o una función y se debe esperar a que Agemap retorne la respuesta o ejecute la solicitud; de ésta manera se pueden hacer programas que tomen información geográfica desde Agemap sin necesidad de salir del entorno de programación conocido.

Capítulo 7

Implementación de la Aplicación

Especificación y Diseño de Sistemas de Software Distribuido de Tiempo Real

Ingeniería del Sistema

Análisis de Requerimientos del Sistema

Diagrama de Contexto de Datos (DCD)

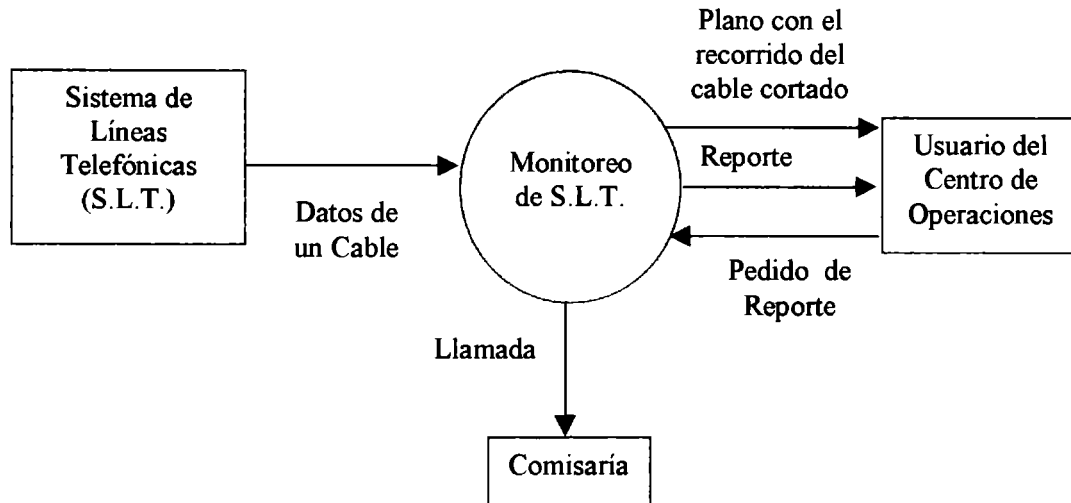
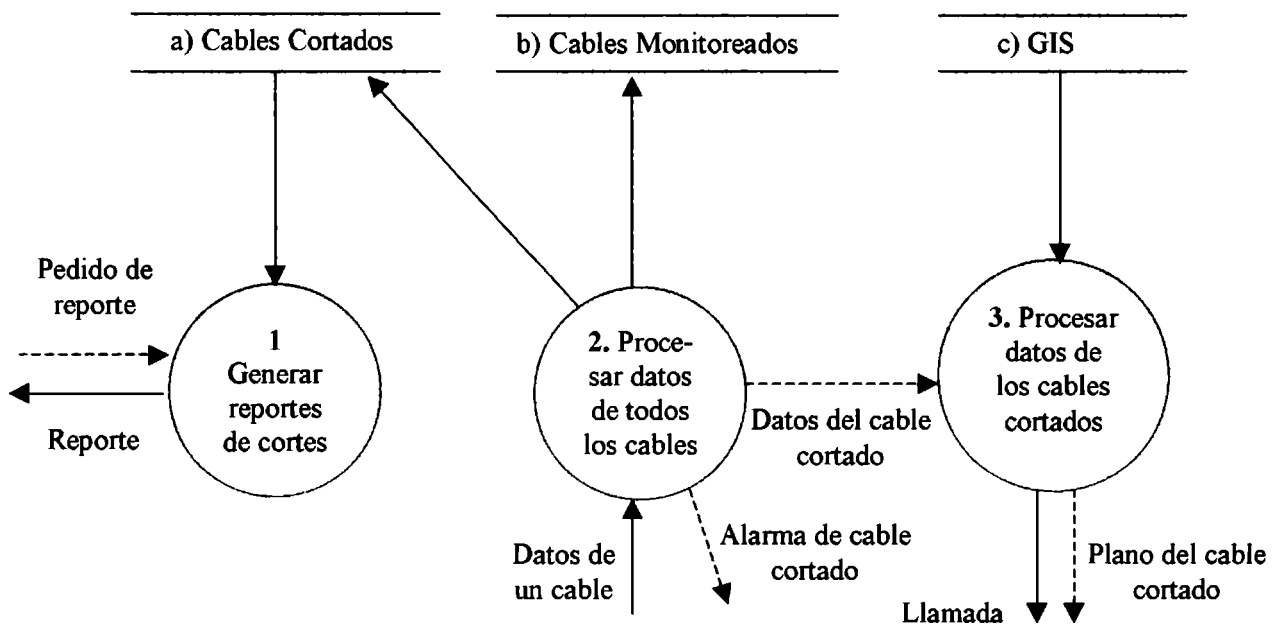
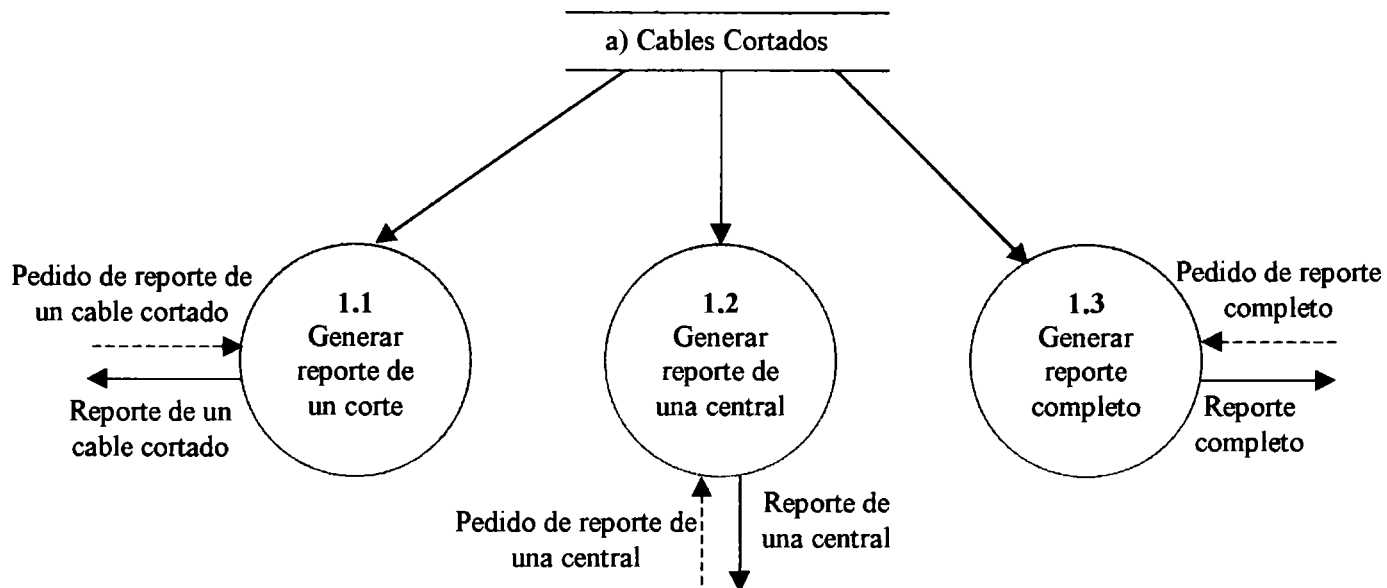


Diagrama de Flujos de Datos (DFD)

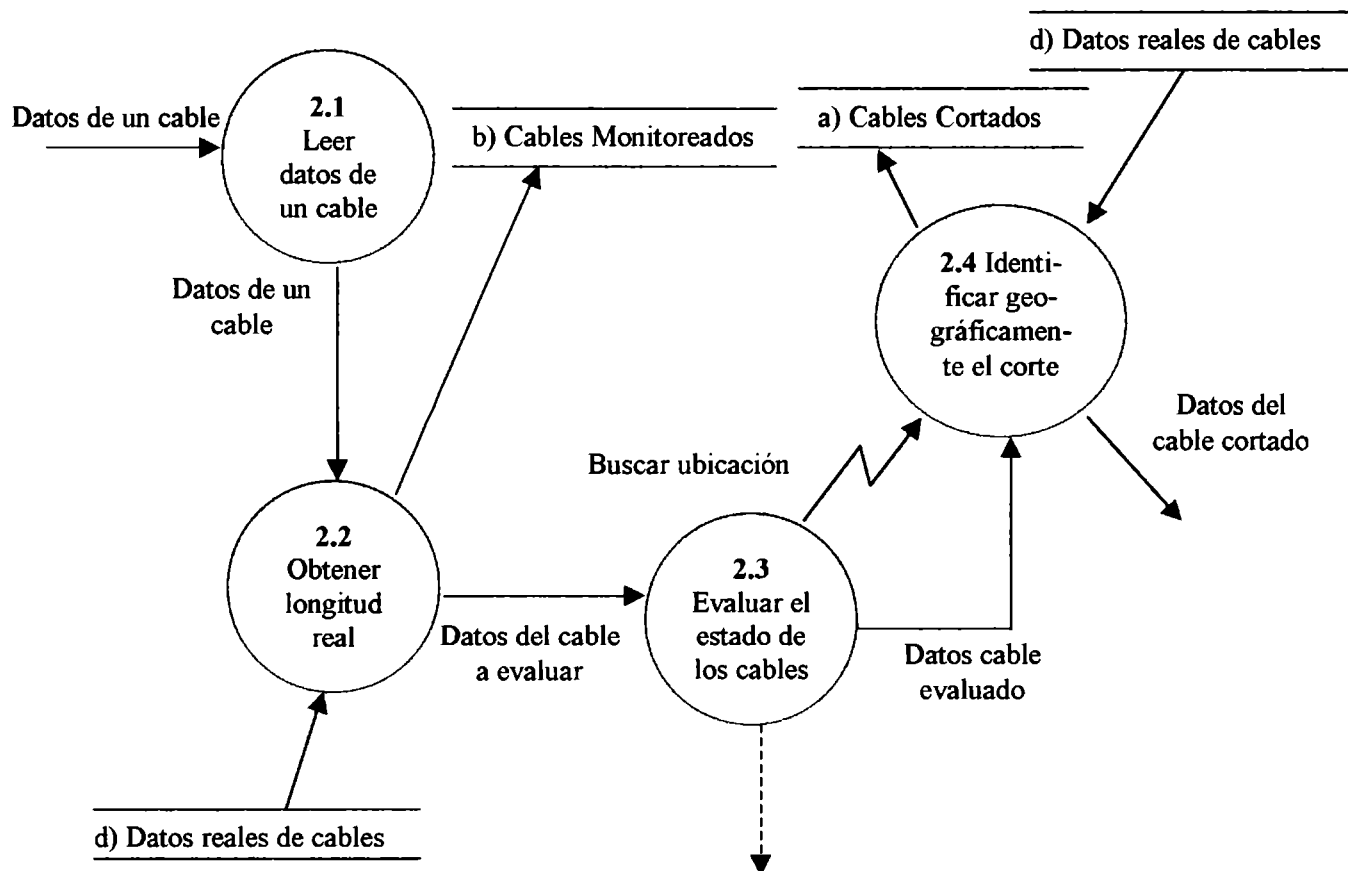
DFD 0



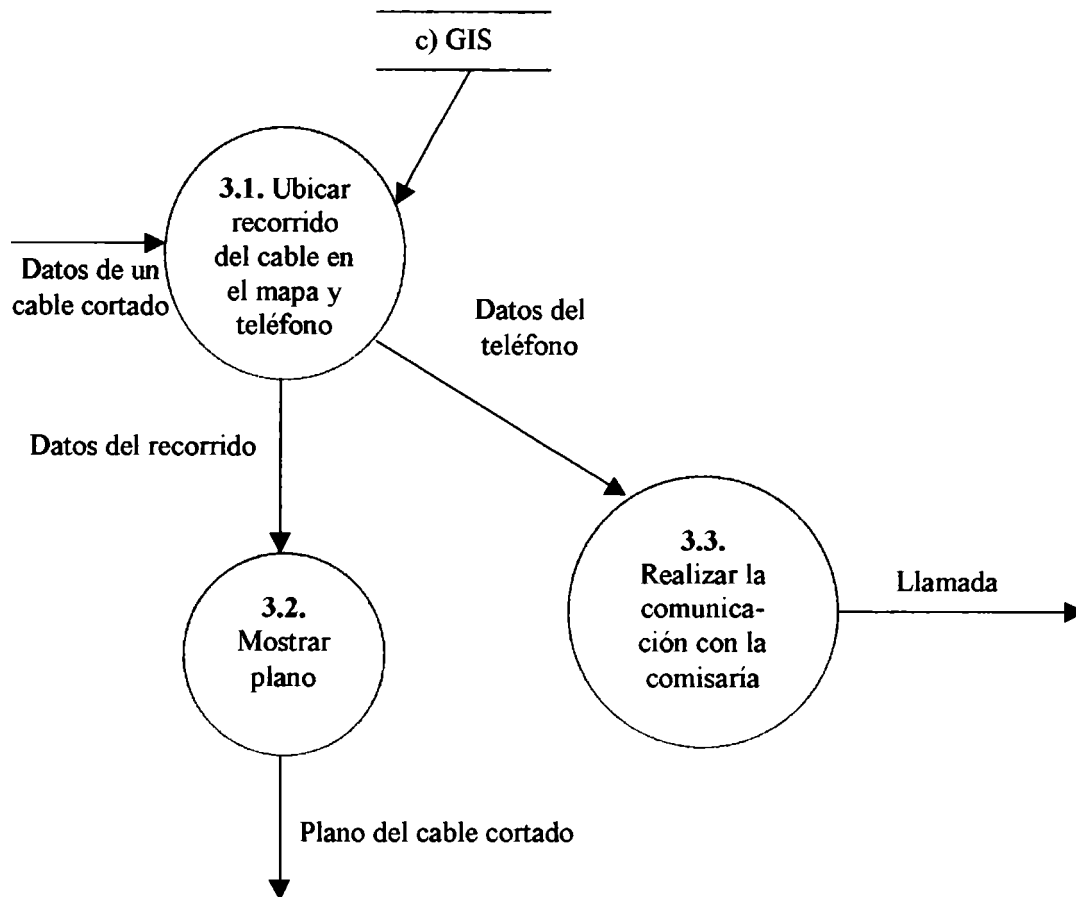
DFD 1



DFD 2



DFD 3



Especificación de procesos (PSpec)

1. Generar reporte de cortes

Este proceso recibe un pedido de reporte del usuario del C.O., lo genera y lo envía

2. Procesar datos de todos los cables

Este proceso estará monitoreando los datos de los cables recibidos por el S.L.T., evalúa el estado y almacena los datos obtenidos en el archivo de cables monitoreados. En caso de detectar un corte envía los datos del cable al proceso **Procesar datos de los cables cortados (3)** y una señal de alarma.

3. Procesar datos de los cables cortados

Este proceso ubica geográficamente el recorrido del cable, lo muestra en el mapa y realiza la comunicación con la comisaría correspondiente.

1.1. Generar reporte de un corte

Este proceso recibe un pedido de reporte de un cable cortado en particular, lo genera y devuelve el reporte generado.

1.2. Generar reporte de una central

Este proceso recibe un pedido de reporte de todos los cables cortados de una central, lo genera y devuelve el reporte generado.

1.3. Generar reporte completo

Este proceso recibe un pedido de reporte de todos los cables cortados de todas las centrales, lo genera y devuelve el reporte generado.

2.1. Leer datos de un cable

Este proceso recibe los datos de los cables monitoreados por el S.L.T.

2.2. Obtener longitud real

Este proceso recibe los datos de un cable cortado y accede al archivo con los datos reales del cable y recupera su longitud original.

2.3. Evaluar el estado de los cables

Este proceso recibe los datos del cable con su longitud real y longitud actual y determina si se produjo un corte en algún lugar de su recorrido. De ser así, activa la alarma en el C.O.

2.4. Identificar geográficamente el corte

Obtiene los datos del recorrido del cable del archivo de los datos reales del cable y almacena los datos en el archivo de cables cortados.

3.1. Ubicar el recorrido del cable en el mapa y teléfono

Este proceso recibe los datos de un cable cortado, ubica el recorrido de cable en el mapa y recupera el teléfono de la comisaría correspondiente. Esto lo hace a través de la Base de Datos Geográfica.

3.2. Mostrar plano

Este proceso se encarga de visualizar el mapa con el recorrido del cable cortado.

3.3. Realizar la comunicación con la comisaría

Este proceso se encarga de realizar la llamada telefónica a la comisaría correspondiente.

Diccionario de datos

ALMACENAMIENTOS

a) Cables cortados

Central + nro. cable + longitud actual + longitud real

b) Cables monitoreados

Central + nro. cable + longitud actual + longitud real

c) GIS

Id.recorrido

d) Datos reales de los cables

Central + nro. cable + longitud real + id. recorrido

Pedido de reporte: [pedido de reporte completo | pedido de reporte de una central | pedido de reporte de un cable cortado]

Reporte: [reporte completo | reporte de una central | reporte de un cable cortado]

Datos de un cable: central + nro. cable + longitud actual

Frecuencia: periódicamente

Datos del cable cortado: central + nro. cable + longitud actual + longitud real + id. recorrido

Llamada: *comunicación telefónica con la comisaría*

Plano del cable cortado: *plano con el recorrido del cable cortado*

Alarma de cable cortado: [On | Off] *se activa en el C.O. cuando la central detecta un corte*

Pedido de reporte de un cable cortado: central + nro. cable

Pedido de reporte de una central: central

Pedido de reporte completo: *pedido de reporte total*

Reporte de un cable cortado: central + nro. cable + longitud real + longitud actual

Reporte de una central: central + (nro. cable + longitud real + longitud actual)⁺

Reporte completo: (central + nro. cable + longitud real + longitud actual)⁺

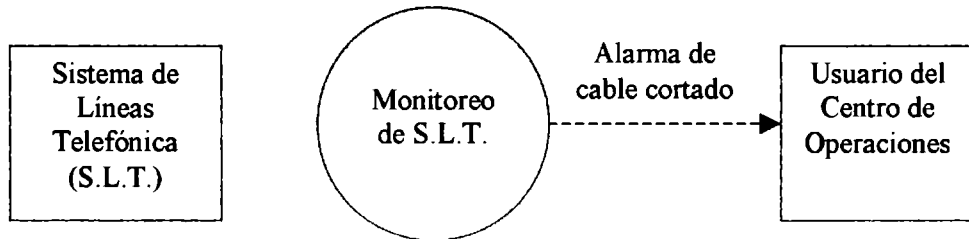
Datos del cable a evaluar: central + nro. cable + longitud actual + longitud real

Datos del cable evaluado: central + nro. cable + longitud actual + longitud real

Datos del recorrido: central + nro. cable + longitud actual + id. recorrido

Datos del teléfono: central + nro. cable + dirección del corte + teléfono

Diagrama de Contexto de Control (CCD)



DCD Y CCD combinado

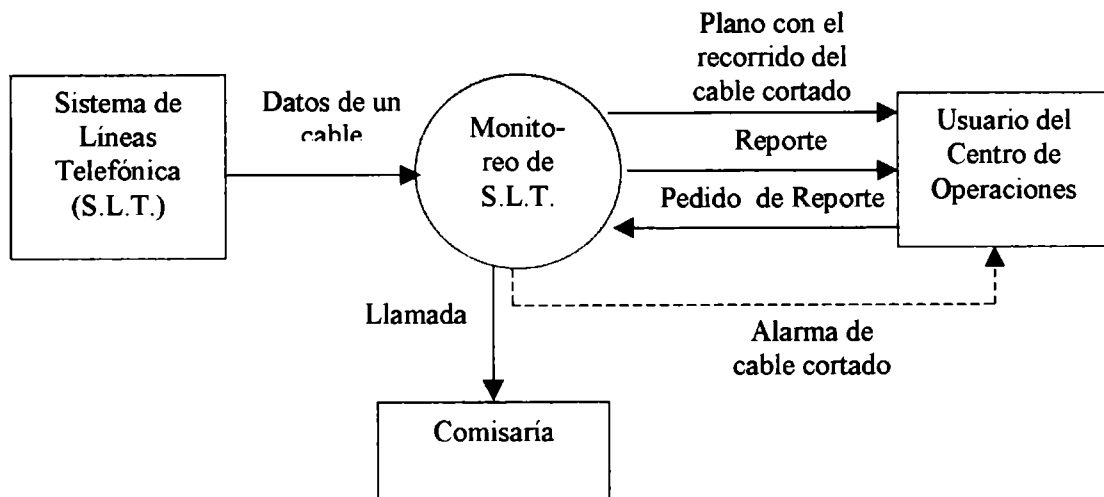
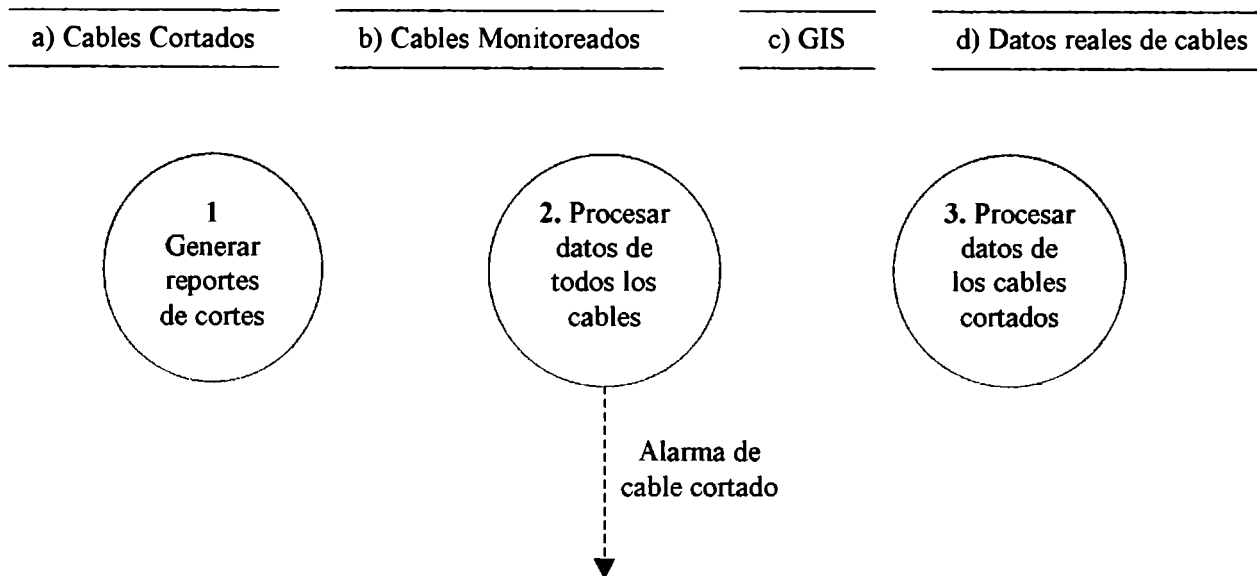


Diagrama de Flujos de Control (CFD)

CFD 0



Especificación de Control (Cspec)

Longitud actual	Sonar alarma	Llamar a la comisaría	Ver recorrido	Almacenar cables cortados	Almacenar cables monitoreados
Longitud actual = longitud real					X
Longitud actual /= longitud real	X	X	X	X	X

Especificación de Tiempos de Respuesta (RTS)

Señal de Entrada Externa	Evento de Entrada	Frecuencia	Señal de Salida Externa	Evento de Salida	Tiempo de Respuesta
Pedido de reporte	Ingresado	Manejado por un evento	Reporte (corte, central o centrales)	Mostrado	Dentro de los 2'
Datos de un cable	Recibido desde el S.L.T.	Periodicamente	Exhibición geográfica si hay corte, alarma y comunicación a la comisaria	Mostrado el plano, alarma activada y comunicación realizada	Dentro de los 3'

Matrices de Requerimientos (RTM)

Necesidades del Usuario (1*) / Componentes del modelo de Requerimientos (2*)

1* \ 2*	Generar reportes 1	Procesar datos de cables 2	Procesar datos de cables cortados 3
1. Solicitar distintos reportes			
1.1. Pedir reporte de un corte	X		
1.2. Pedir reporte de los cortes de una central	X		
1.3. Pedir reporte de todas las centrales	X		

///

///

2. Interactuar con el S.L.T.		X	
3. Detectar el corte del cable		X	
3.1. Sonar alarma		X	
4. Cable cortado			
4.1. Obtener datos del cable			X
4.2. Sonar alarma			X
4.3. Mostrar recorrido			X
4.4. Realizar comunicación con la comisaría			X

Necesidades del Usuario (1*) / Componentes del modelo de Requerimientos (2*)

<div>2*</div> <div>1*</div>	Generar reporte de un corte 1.1	Generar reporte de una central 1.2	Generar reporte completo 1.3
1.1. Pedir reporte de un corte	X		
1.2. Pedir reporte de los cortes de una central		X	
1.3. Pedir reporte de todas las centrales			X

Necesidades del Usuario (1*) / Componentes del modelo de Requerimientos (2*)

1* \ 2*	Leer datos de un cable 2.1	Obtener longitud real 2.2	Evaluar estado del cable 2.3	Ubicar geográficamente el corte 2.4
2. Interactuar con el S.L.T.	X			
3. Detectar el corte de un cable		X	X	
4. Cable cortado				
4.1. Obtener datos del cable				X
4.2. Sonar alarma			X	
4.3. Mostrar recorrido				X
4.4. Realizar comunicación con la comisaría				X

Necesidades del Usuario (1*) / Componentes del modelo de Requerimientos (2*)

1* \ 2*	Ubicar el recorrido del cable en el mapa y teléfono 3.1	Mostrar plano 3.2	Realizar comunicación con la comisaría 3.3
4. Cable cortado	X		
4.1. Ver recorrido		X	
4.2. Llamar por teléfono a la comisaría			X

Matriz Acción / Almacenamiento de Datos

Alm. de Datos \ Acción	Cables cortados	Cables monito-reados	GIS	Datos reales de cables
Generar reporte de un corte	U			
Generar reporte de una Central	U			
Generar reporte completo	U			
Procesar datos de todos los cables	S	S,U		U
Procesar tareas referentes a un cable cortado			U	

Lista de Eventos

1. Pedido de Reporte de un corte:

Cuando se recibe un PEDIDO DE REPORTE DE UN CORTE desde el usuario del Centro de Operaciones, el sistema lo genera.

2. Pedido de Reporte Completo sobre los cortes de una central:

Cuando se recibe un PEDIDO DE REPORTE COMPLETO DE UNA LAS CENTRALES desde el usuario del Centro de Operaciones, el sistema lo genera.

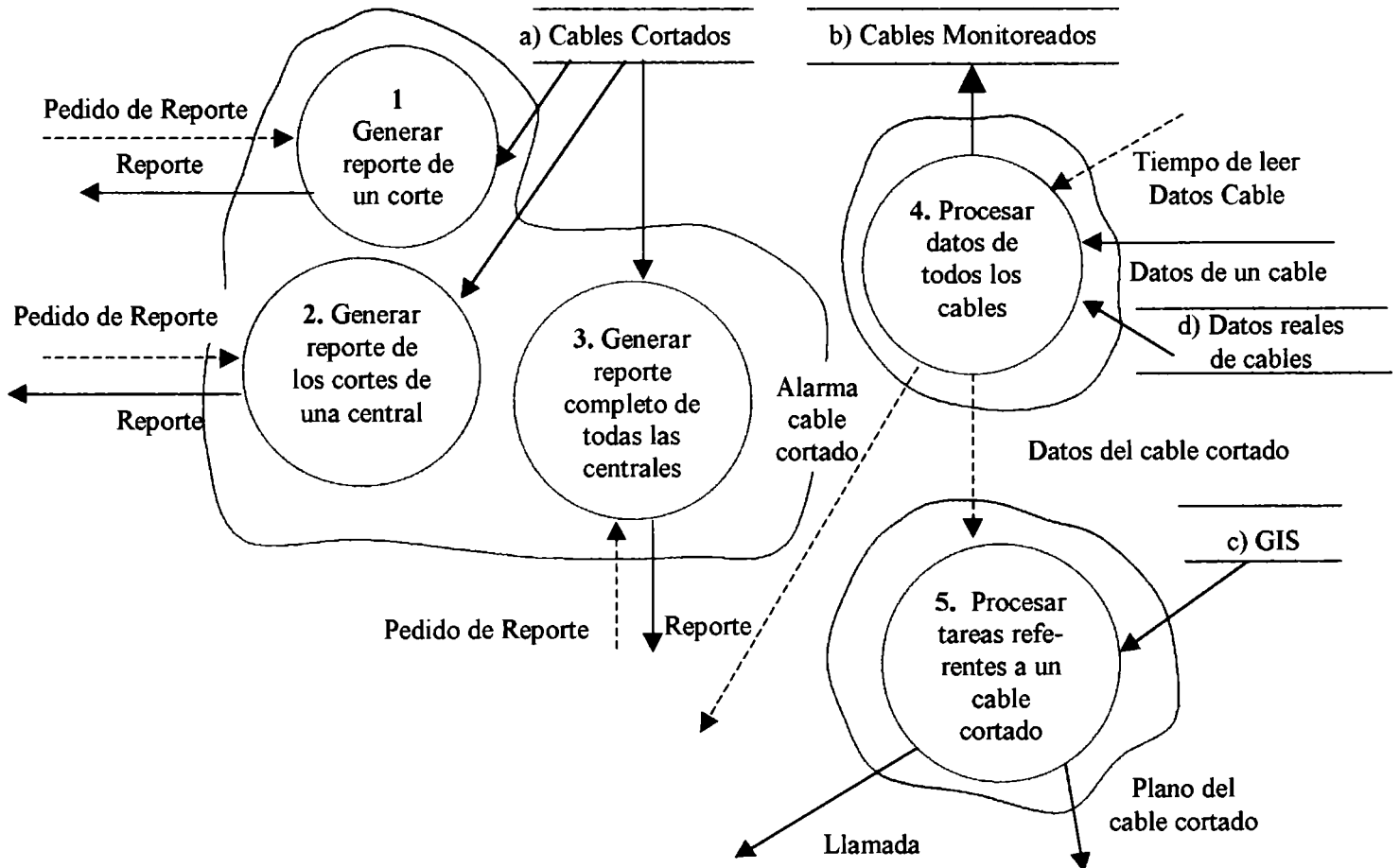
3. Pedido de Reporte Completo sobre todos los cortes de todas las centrales:

Cuando se recibe un PEDIDO DE REPORTE COMPLETO DE TODAS LAS CENTRALES desde el usuario del Centro de Operaciones, el sistema lo genera.

4. Información de un cable:

Cuando se reciben los DATOS DE UN CABLE desde un SUPERVISOR DE LINEAS TELEFONICAS, hacer los cálculos necesarios y guardar todo en CABLES MONITOREADOS. Si se detecta que el cable está cortado, mostrar el recorrido del mismo y la ubicación geográfica del corte por medio de un GIS en el Centro de Operaciones y guardar la información en CABLES CORTADOS, realizando la comunicación a la comisaria correspondiente.

Diagrama de Evento – Acción



Diseño del Sistema

Alternativas de Diseño

La alternativa que se presenta para el diseño es la siguiente:

Cada Central estará conectada con el S.L.T. y con el Centro de Operaciones a través de una P.C.

Las funciones de cada Central son:

1. Tomar los datos de los cables monitoreados por el S.L.T. y evaluarlos para determinar si hay alguno cortado.
2. Enviar al centro de operaciones los datos correspondientes al cable cortado y enviar señal de alarma.

En el Centro de Operaciones habrá una P.C. conectada con cada Central y con el G.I.S. que tiene la información del tendido de cables y las comisarias.

Las funciones del Centro de Operaciones son:

1. Consultar el G.I.S. determinando la ubicación del corte.
2. Mostrar el plano con el recorrido del cable cortado.
3. Determinar la comisaría a la que pertenece la zona del corte y realizar la comunicación con la misma.

Diagrama de Contexto de la Arquitectura (ACD)

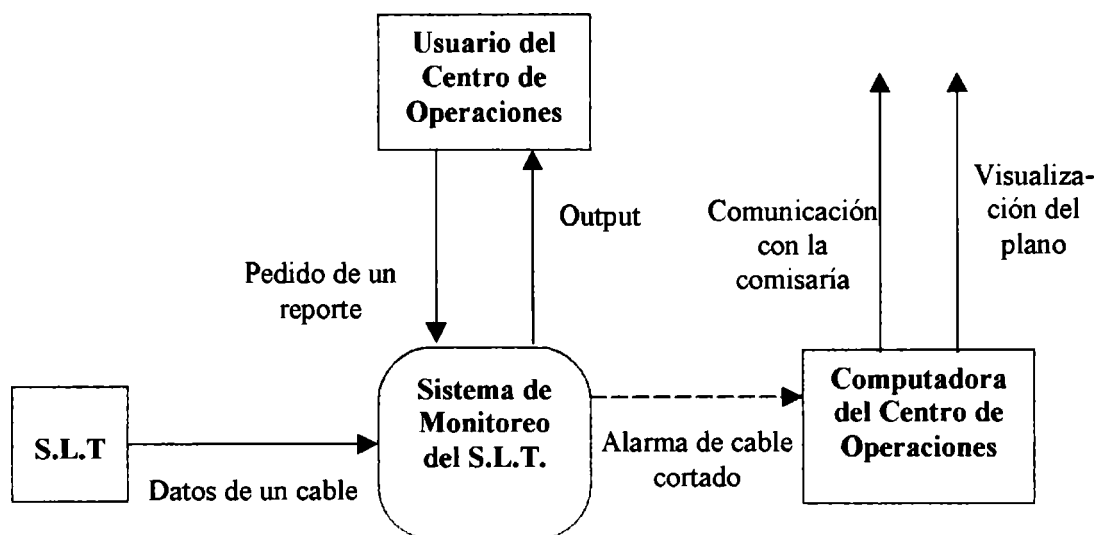
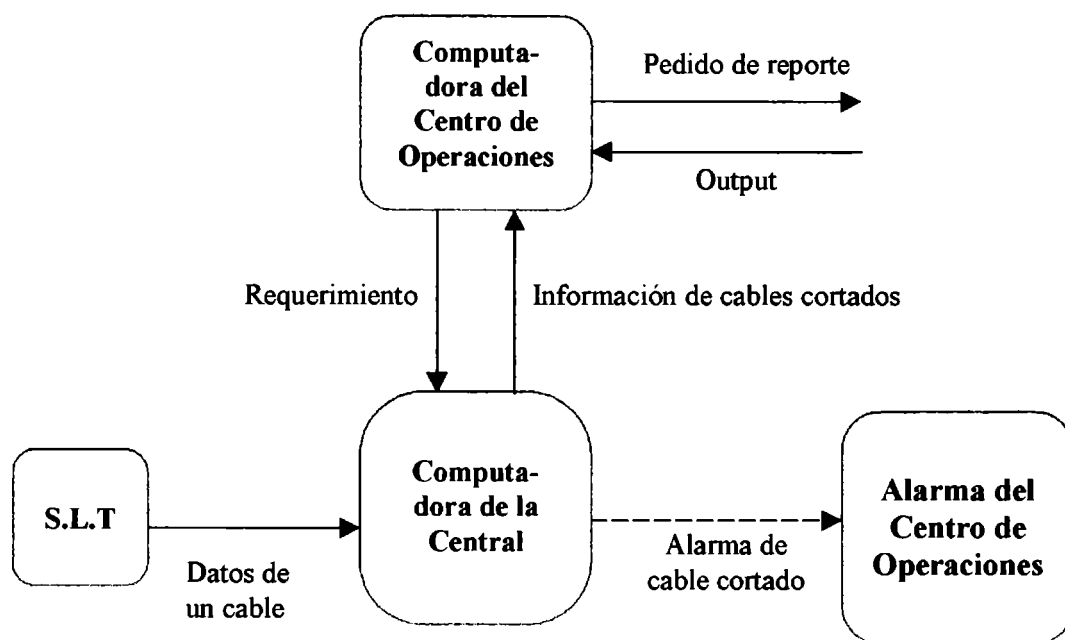


Diagrama de Flujo de la Arquitectura (AFD)

AFD 0



Especificación de los módulos de la Arquitectura (Mspec)

1. Computadora del C.O.

La C.C.O. incluirá el software necesario para:

- Obtener los datos de los cables cortados de las centrales.
- Ingresar pedidos de reportes y la generación de los mismos con los displays correspondientes.
- Realizar la llamada telefónica a la comisaría.
- Realizar la visualización del recorrido de un cable cortado mediante un GIS.

2. Computadora de la Central.

La C.C. incluirá el software necesario para:

- Obtener los datos de los cables monitoreados por el S.L.T.
- Evaluar si se produjo algún corte.
- Transmitir a la C.C.O. la información necesaria si se detecta un corte.
- Registrar en archivos de datos de cables monitoreados y cortados.

3. S.L.T.

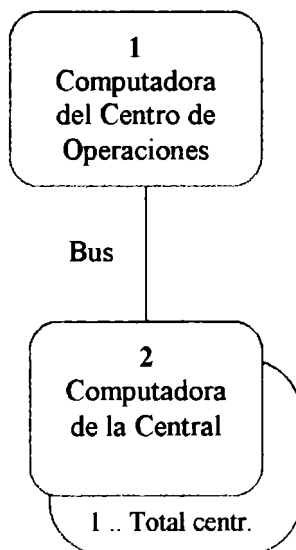
Sistema monitorador de cables telefónicos.

4. Alarma del Centro de Operaciones

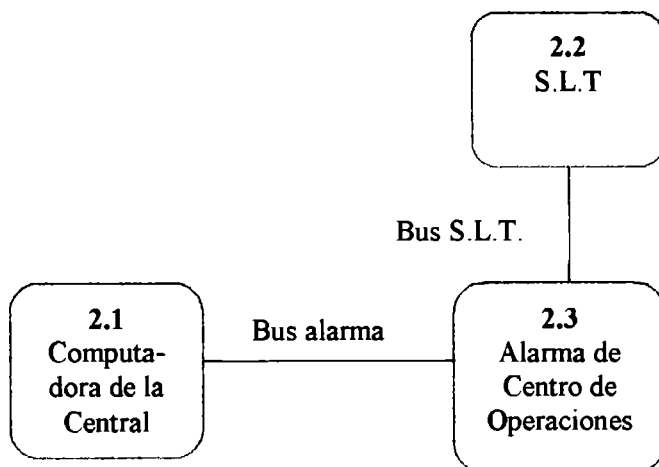
Es el hardware que emitirá un sonido de alarma en el C.O. si la central detecta un corte de un cable.

Diagrama Interconectado de la Arquitectura (AID)

AID 0



AID 2



Especificación de la Interconexión de la arquitectura

AIS 0

Bus: bus usado para transmitir información entre la central y el C.O.

AIS 2

Bus SLT: bus usado para transmitir información entre el S.L.T. y la central.

Bus alarma: bus usado para activar la alarma.

Diccionario de la Arquitectura

Pedido de reporte: [pedido de reporte completo | pedido de reporte de una central | pedido de reporte de un cable cortado]

Output: [reporte completo | reporte de una central | reporte de un cable cortado | mapa | llamada]

Datos de un cable: central + nro. cable + longitud actual

Información de cables cortados: central + nro. cable + longitud actual + longitud real + id. recorrido

Reconocimiento: *señal de aceptación de los datos recibidos*

Alarma de cable cortado: [On | Off] *sonido de alarma que se activa en el C.O. cuando la central detecta un corte*

Alocación de Tiempos de Respuesta (RTA)

Evento de Entrada	Evento de Salida	Computadora de la Central	Computadora del Centro de Operaciones	Tiempo total de Respuesta
Pedido de reporte de un corte en particular	Imprimir los datos del cable cortado			Dentro de 1'
Pedido de reporte de los cortes de una central	Imprimir los datos de todos los cortes de una central			Dentro de los 2'
Pedido de reporte de los cortes todas las centrales	Imprimir los datos de todos los cortes de todas las centrales			Dentro de los 2,5'
Datos de un cable recibidos desde el S.L.T.	Mostrar el plano con el recorrido, si el cable está cortado	1'	2'	Dentro de los 3'
	Realizar la comunicación con la comisaría		2'	Dentro de los 3'

Matriz de Requerimientos (RTM)

Componentes del modelo de Requerimientos (1*)/Componentes del modelo de Arquitectura(2*)

1* \ 2*	Computadora del Centro de Operaciones 1	Computadora de la Central 2
1.1. Pedir reporte de un corte	X	
1.2. Pedir reporte de los cortes de una central	X	
1.3. Pedir reporte de todas las centrales	X	

///

///

2.1. Leer datos de un cable		X
2.2. Calcular longitud		X
2.3. Evaluar el estado de los cables		X
2.4. Identificación geográfica del recorrido		X
3.1. Ubicar el recorrido del cable en el mapa y teléfono	X	
3.2. Mostrar plano	X	
3.3. Realizar la comunicación con la comisaría	X	

Ingeniería del Software

Análisis de Requerimientos del Software

Diagrama de Flujo mejorado para la C.C. (E.F.D.)

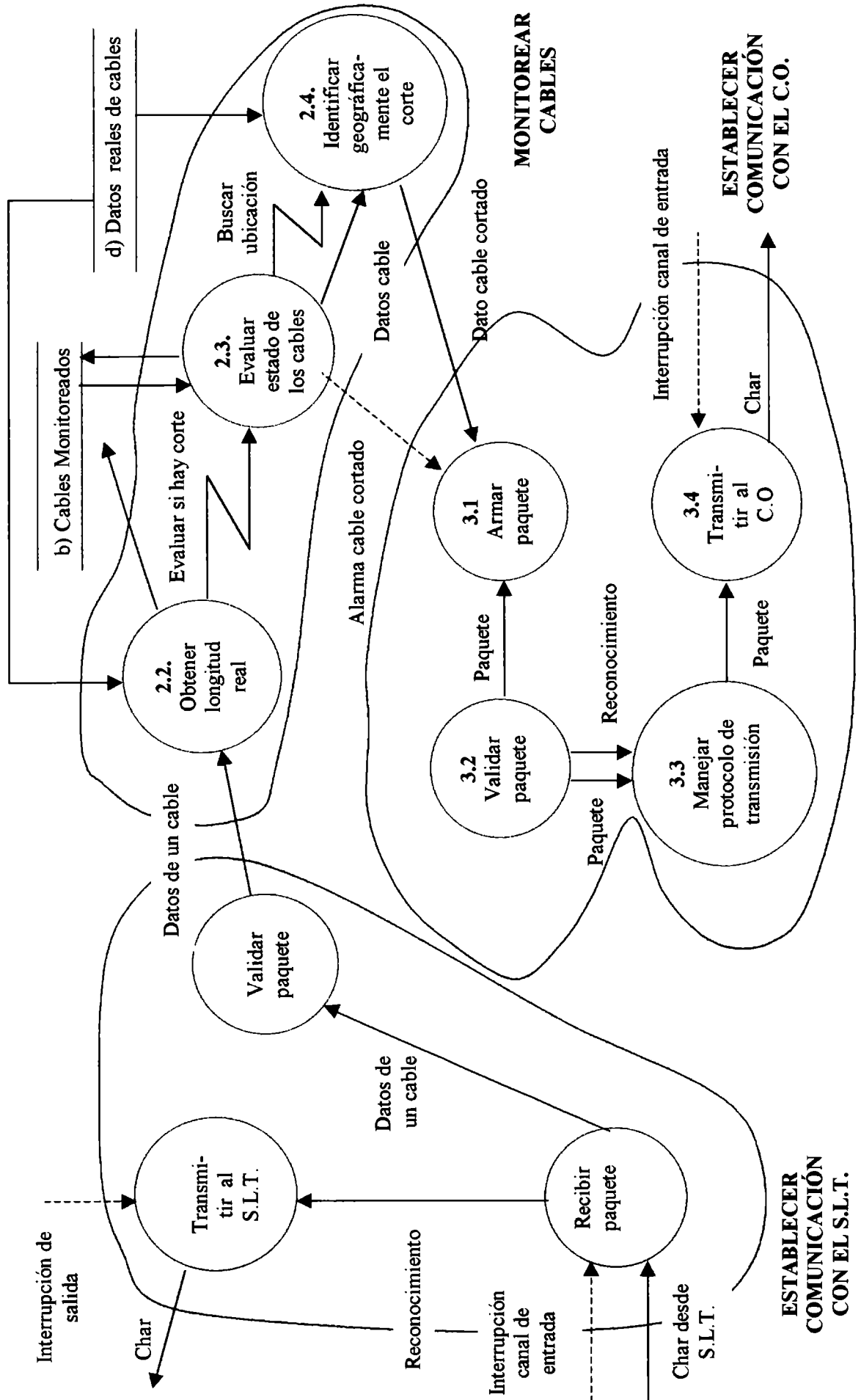


Diagrama de Contexto de Datos /Control para la Computadora de la Central

DCD/CCD para C.C.

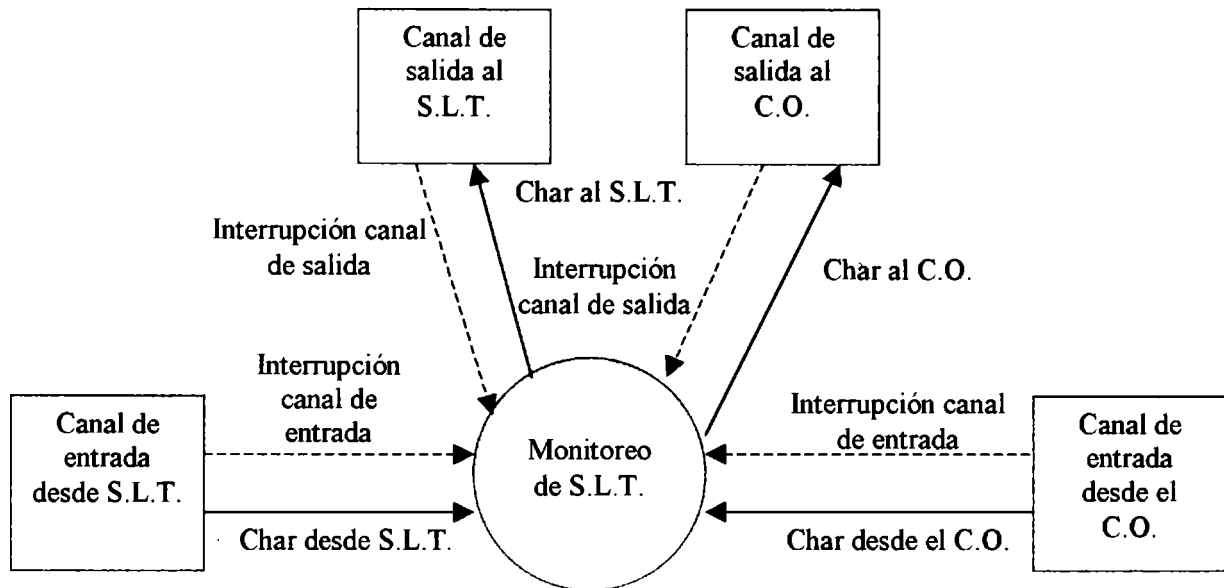
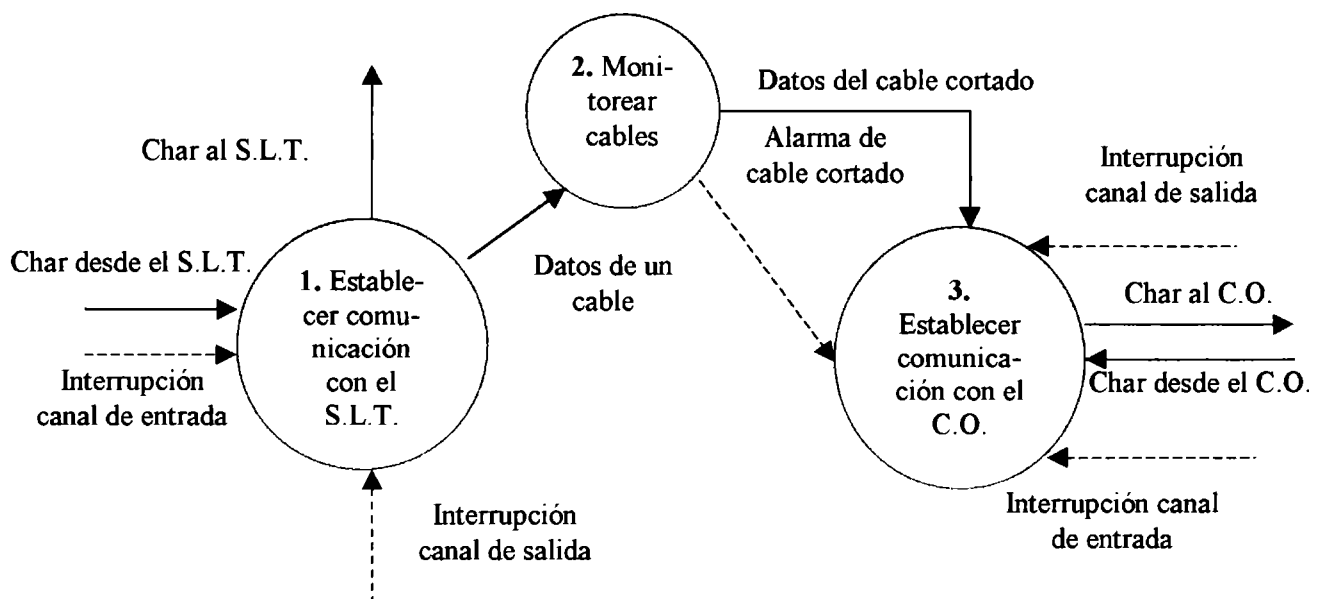
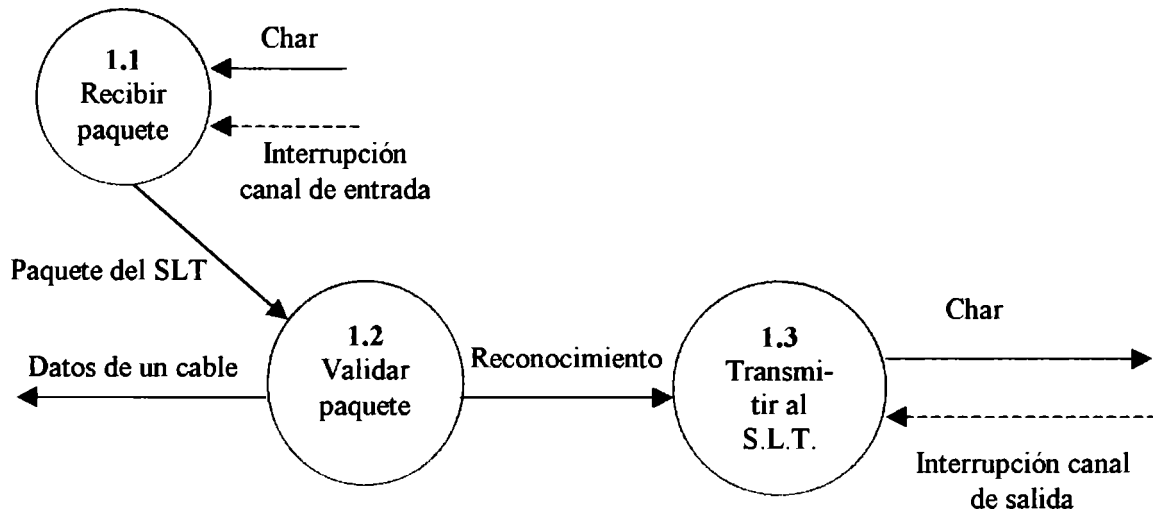


Diagrama de Flujos de Datos / Diagrama de Flujos de Control para la Computadora de la Central

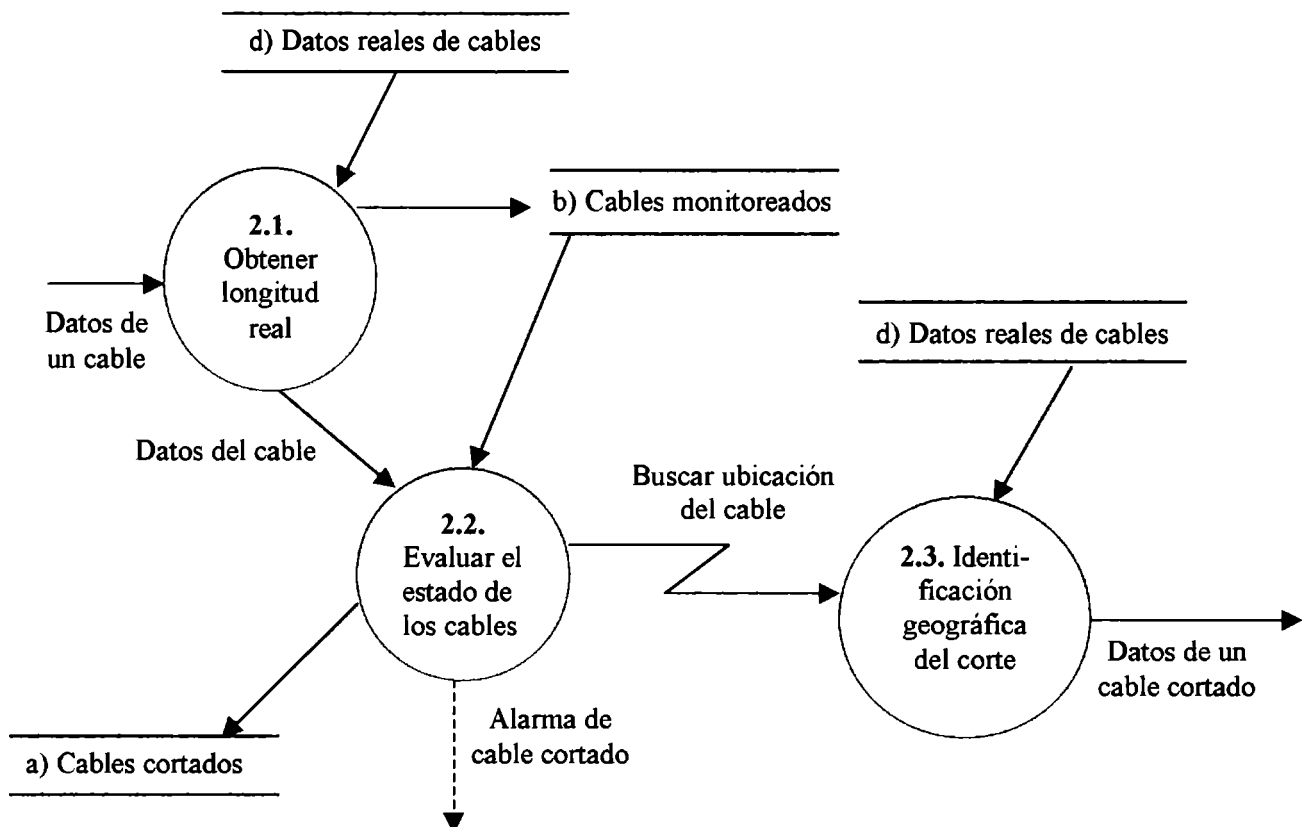
DFD / CFD 0



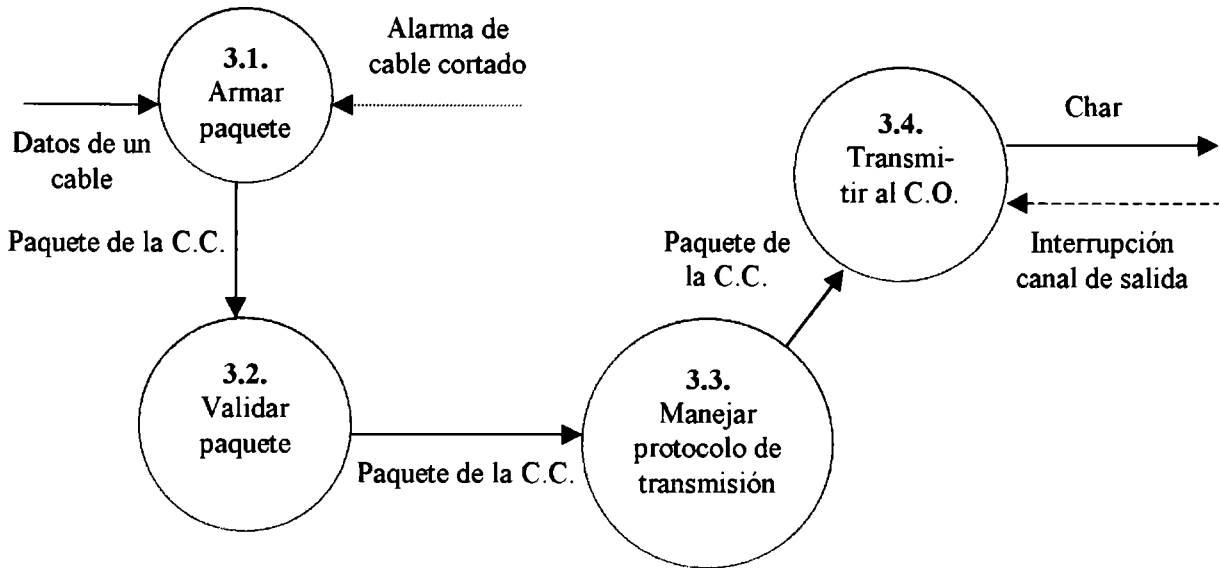
DFD 1 para la computadora de la Central



DFD 2 para la computadora de la Central



DFD 3 para la computadora de la Central



Especificación de Procesos (PSpec para la C.C.)

1. Establecer comunicación con el S.L.T.

Establece la comunicación y recibe un paquete.

2. Monitorear cables

Este proceso estará monitoreando los datos de los cables recibidos por el S.L.T., evalúa el estado y almacena los datos obtenidos en el archivo de cables monitoreados. En caso de detectar un corte envía los datos del cable al proceso **Procesar datos de los cables cortados (3)** y una señal de alarma.

3. Establecer comunicación con el C.O.

Establece la comunicación con el C.O. para transmitir y recibir datos.

1.1. Recibir paquete

Proceso encargado de tomar datos del buffer.

1.2. Validar paquete

Evalúa el paquete y determina si es válido enviando un reconocimiento al S.L.T.

1.3. Transmitir al S.L.T.

Encargado de transmitir datos.

2.1. Obtener longitud real

Este proceso recibe los datos de un cable cortado y accede al archivo con los datos reales del cable y recupera su longitud original.

2.2. Evalúa el estado de los cables

Este proceso recibe los datos del cable con su longitud real y longitud actual y determina si se produjo un corte en algún lugar de su recorrido. De ser así, activa la alarma en el C.O.

2.3. Identificación geográfica del corte

Obtiene los datos del recorrido del cable del archivo de los datos reales del cable y almacena los datos en el archivo de cables cortados.

3.1. Armar paquete

Proceso que arma un paquete con los datos de un cable y la alarma para transmitir al C.O.

3.2. Validar paquete

Valida el paquete recibido de **Armar paquete** y envía el paquete para establecer los protocolos.

3.3. Manejar protocolo de transmisión

Arma el paquete de acuerdo al protocolo requerido.

3.4. Transmitir al C.O.

Transmitir al C.O.

Diccionario de datos para la C.C.

Char desde el SLT: *carácter de paquete de datos*

Char al SLT: *caracter de reconocimiento*

Interrupción canal de entrada: *interrupción de hardware indicando que el char desde el SLT ha sido puesto en el buffer*

Interrupción canal de salida: *indicación de hardware indicando que un char ha sido puesto en el buffer*

Char al C.O.: *carácter de paquete de datos*

Char desde el C.O.: *caracter de reconocimiento*

Interrupción canal de entrada: *interrupción de hardware indicando que el char desde el SLT ha sido puesto en el buffer*

Interrupción canal de salida: *indicación de hardware indicando que un char ha sido puesto en el buffer*

Paquete del SLT: central + nro. cable + longitud actual *datos recibidos del SLT*

Reconocimiento: *señal de aceptación del paquete ingresado*

Buscar ubicación del cable: *proceso activador*

Paquete de la C.C.: *datos recibidos de la C.C.*

Datos de un cable: central + nro. cable + longitud actual

Frecuencia: periódicamente

Datos del cable cortado: central + nro. cable + longitud actual + longitud real + id. recorrido

Alarma de cable cortado: [On | Off] *se activa en el C.O. cuando la central detecta un corte*

Especificación de Tiempos Externos (RTS para la C.C.)

Señal de Entrada Externa	Evento de Entrada	Frecuencia	Señal de Salida Externa	Evento de Salida	Tiempo de Respuesta
Datos de un cable	Recibido desde el S.L.T.	Periódicamente	Paquete con datos de un cable cortado	Enviado al C.O.	24 seg. aproximadamente
Reconocimiento	Recibido desde el C.O.	Manejado por un evento	Paq. con datos de un cable cortado o con señal de control	Enviado al C.O. si está disponible	Despreciable
Interrupción del canal de entrada	Recibido desde el S.L.T.	Periódicamente	Ninguna	Tomar un caracter del buffer	Despreciable
Interrupción del canal de entrada	Recibido desde el C.O.	Manejado por un evento	Ninguna	Tomar un caracter del buffer	Despreciable

///

///

Interrupción del canal de entrada	Recibido desde el C.O.	Manejado por un evento	Ninguna	Tomar un caracter del buffer	Despreciable
Interrupción del canal de salida	Recibido desde el S.L.T.	Manejado por un evento	Ninguna	Poner próx. caracter en el buffer	Despreciable
Interrupción del canal de salida	Recibido desde el C.O.	Manejado por un evento	Ninguna	Poner próx. caracter en el buffer	Despreciable

Matrices de Requerimientos (RTM para la C.C.)

Requerimientos del Sistema (1*) / Requerimientos del Software (2*)

1* \ 2*	Establecer comunicación con el S.L.T. 1	Monitorear Cables 2	Establecer comunicación con la C.C.O. 3
2.1. Obtener longitud real del cable		X	
2.2. Evaluar estado de los cables		X	
2.3. Identificación geográfica del corte		X	
1.1. Recibir paquete	X		
1.2. Validar paquete	X		
1.3. Transmitir al S.L.T.	X		

///

///

3.1. Armar paquete			X
3.2. Validar paquete			X
3.3. Manejar protocolo de transmisión			X
3.4. Transmitir al C.O.			X

Requerimientos del Sistema (1*) / Requerimientos del Software (2*)

1* \ 2*	Recibir paquete 1.1.	Validar paquete 1.2.	Transmitir al S.L.T. 1.3.
1.1. Recibir paquete	X		
1.2. Validar paquete		X	
1.3. Transmitir al S.L.T.			X

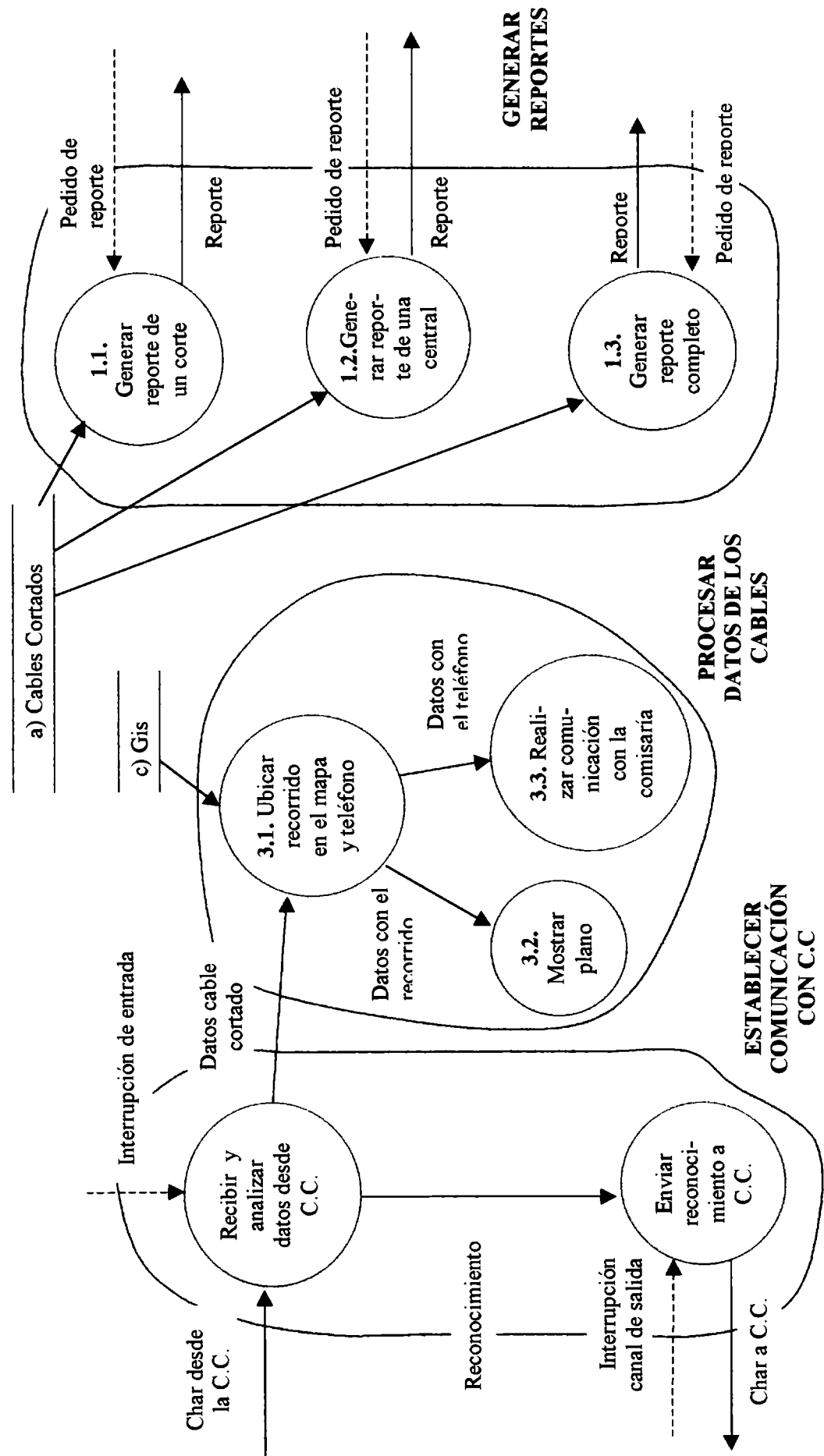
Requerimientos del Sistema (1*) / Requerimientos del Software (2*)

1* \ 2*	Obtener longitud real 2.1.	Evaluar el estado de los cables 2.2.	Identificar geo-gráficamente el corte 2.3
2.1. Obtener la longitud real del cable	X		
2.2. Evaluar el estado de los cables		X	
2.3. Identificación geográfica del corte			X

Requerimientos del Sistema (1*) / Requerimientos del Software (2*)

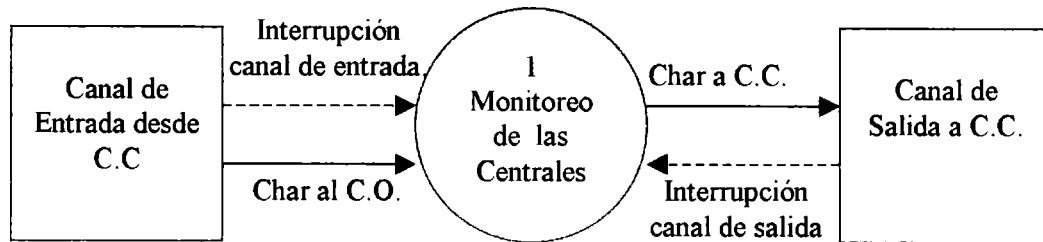
<div>2*</div> <div>1*</div>	Recibir paquete 3.1.	Validar paquete 3.2.	Manejar protocolo de transmisión 3.3.	Transmitir al C.O. 3.4
3.1. Armar paquete	X			
3.2. Validar paquete		X		
3.3. Manejar protocolo de transmisión			X	
3.4. Transmitir al C.O.				X

Diagrama de Flujo mejorado para la computadora del C.O. (E.F.D.)



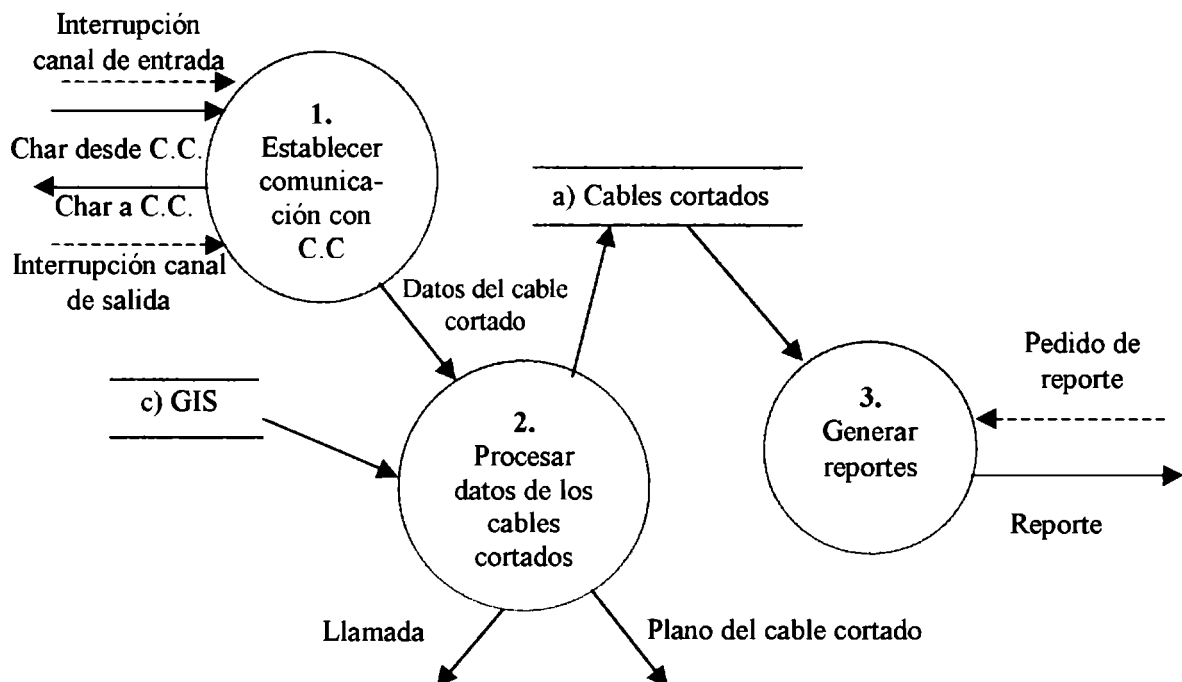
**Diagrama de Contexto de Datos /Control
para la Computadora del Centro de Operaciones.**

DCD/CCD para C.C.O.

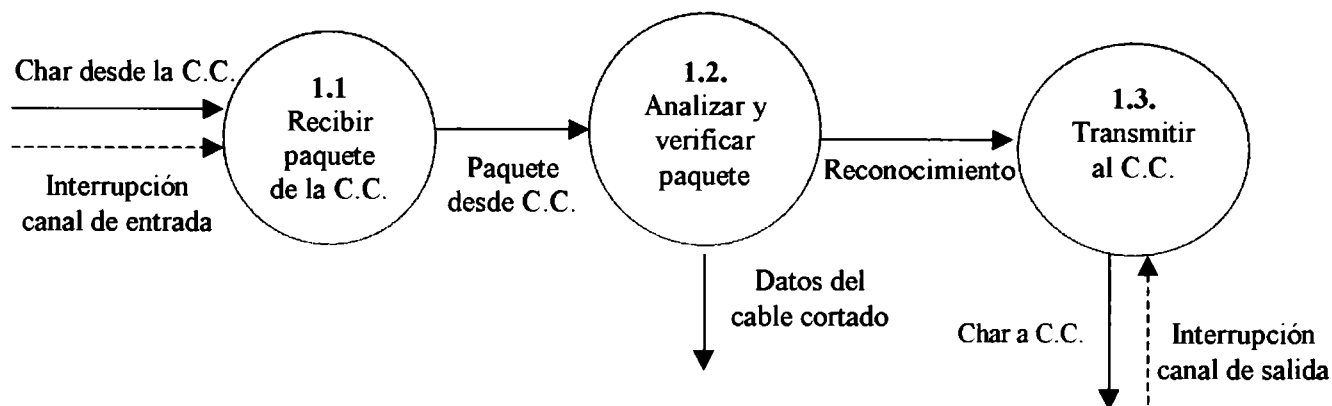


**Diagrama de Flujos de Datos / Diagrama de Flujos de Control para la
computadora del Centro de Operaciones**

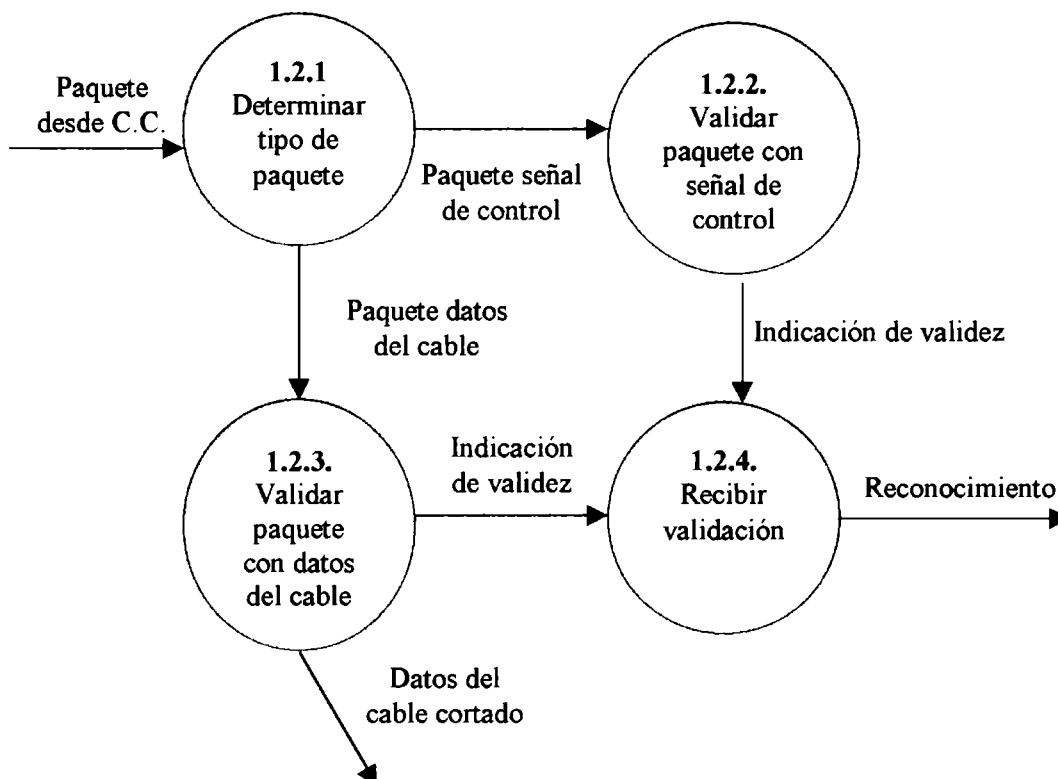
DFD/CFD 0



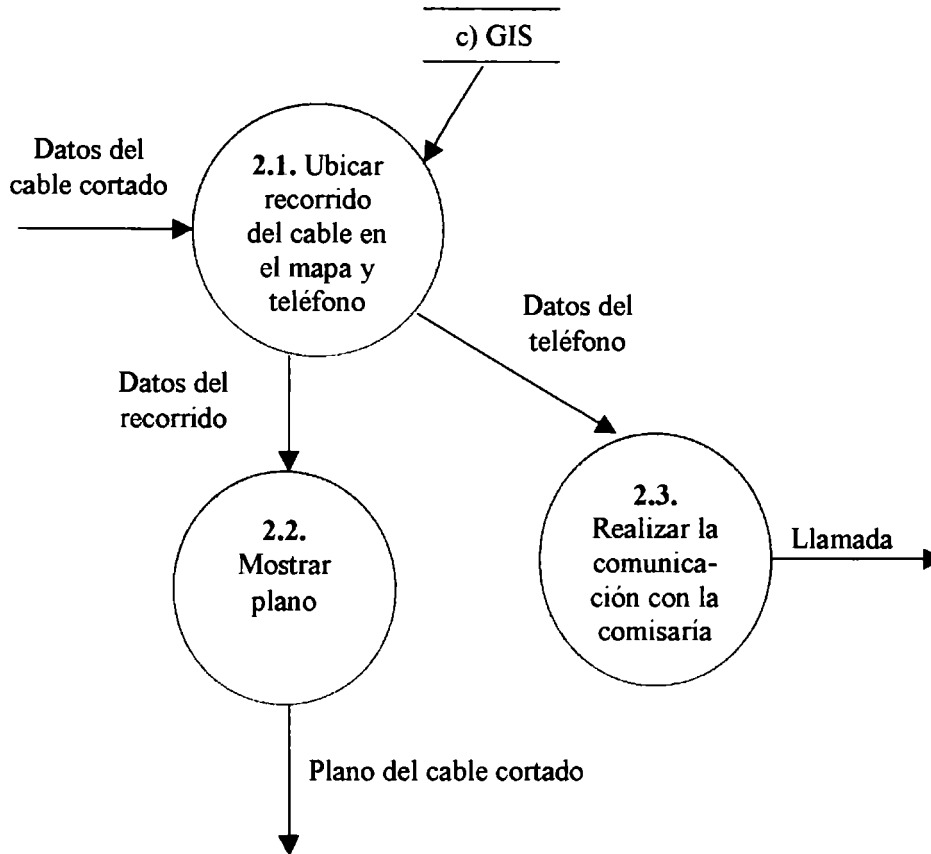
DFD 1 para la computadora del C.O.



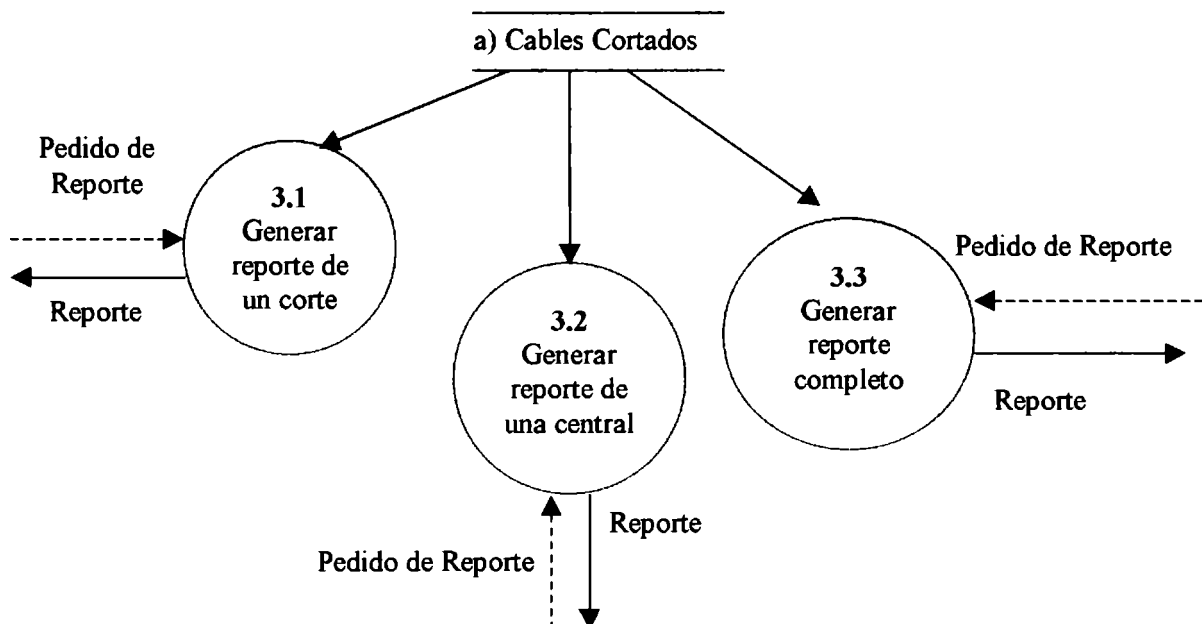
DFD 1.2 para la computadora del C.O.



DFD 2 para la computadora del C.O.



DFD 3 para la computadora del C.O.



Especificación de Procesos (PSpec para la C.C.O.)

1. Establecer comunicación con C.C.

Establece la comunicación con la C.C. y recibe los datos de los cables cortados.

2. Procesar datos de los cables cortados

Este proceso ubica geográficamente el recorrido del cable, lo muestra en el mapa y realiza la comunicación con la comisaría correspondiente.

3. Generar reporte de cortes

Este proceso recibe un pedido de reporte del usuario del C.O., lo genera y lo envía

1.1. Recibir paquete de la C.C.

Encargado de tomar datos del buffer.

1.2. Analizar y verificar paquete

Recibe un paquete, lo analiza y verifica enviando un reconocimiento a la C.C.

1.3. Transmitir a la C.C.

Encargado de transmitir datos.

1.2.1. Determinar el tipo de paquete

Determinar si es paquete de datos o de control.

1.2.2. Validar paquete con señal de control

Valida paquete y envía señal de validez.

1.2.3. Validar paquete con datos

Valida paquete y envía señal de validez.

1.2.4. Recibir validación

Recibe las indicaciones de validez y envía reconocimiento.

2.1. Ubicar el recorrido del cable en el mapa y teléfono

Este proceso recibe los datos de un cable cortado, ubica el recorrido de cable en el mapa y recupera el teléfono de la comisaría correspondiente. Esto lo hace a través de la Base de Datos Geográfica.

2.2. Mostrar plano

Este proceso se encarga de visualizar el mapa con el recorrido del cable cortado.

2.3. Realizar la comunicación con la comisaría

Este proceso se encarga de realizar la llamada telefónica a la comisaría correspondiente.

3.1. Generar reporte de un corte

Este proceso recibe un pedido de reporte de un cable cortado en particular, lo genera y devuelve el reporte generado.

3.2. Generar reporte de una central

Este proceso recibe un pedido de reporte de todos los cables cortados de una central, lo genera y devuelve el reporte generado.

3.3. Generar reporte completo

Este proceso recibe un pedido de reporte de todos los cables cortados de todas las centrales, lo genera y devuelve el reporte generado.

Diccionario de datos para la C.C.O.

Char desde C.C.: *carácter de paquete de datos*

Char a C.C.: *caracter de reconocimiento*

Interrupción canal de entrada: *interrupción de hardware indicando que el char desde la C.C. ha sido puesto en el buffer*

Interrupción canal de salida: *indicación de hardware indicando que un char ha sido puesto en el buffer*

Pedido de reporte: [pedido de reporte completo | pedido de reporte de una central | pedido de reporte de un cable cortado]

Reporte: [reporte completo | reporte de una central | reporte de un cable cortado]

Datos del cable cortado: central + nro. cable + longitud actual + longitud real + id. recorrido

Llamada: *comunicación telefónica con la comisaría*

Plano del cable cortado: *plano con el recorrido del cable cortado*

Pedido de reporte de un cable cortado: central + nro. cable

Pedido de reporte de una central: central

Pedido de reporte completo: *pedido de reporte total*

Reporte de un cable cortado: central + nro. cable + longitud real + longitud actual

Reporte de una central: central + (nro. cable + longitud real + longitud actual)⁺

Reporte completo: (central + nro. cable + longitud real + longitud actual)⁺

Datos del recorrido: central + nro. cable + longitud actual + id. recorrido

Datos del teléfono: central + nro. cable + dirección del corte + teléfono

Paquete desde C.C.:

Reconocimiento:

Paquete señal de control:

Paquete datos del cable:

Indicación de validez: [Si | No]

Especificación de Tiempos Externos (RTS para la C.C.O.)

Señal de Entrada Externa	Evento de Entrada	Frecuencia	Señal de Salida Externa	Evento de Salida	Tiempo de Respuesta
Datos de un cable	Recibido desde la central	Manejado por un evento	Paquete con datos de un cable cortado y la señal de alarma	Llamada a la comisaría y visualización geográfica del corte	21'', aproximadamente
Interrupción del canal de entrada	Recibido desde la central.	Manejado por un evento	Ninguna	Tomar un caracter del buffer	Despreciable
Interrupción del canal de salida	Recibido desde la central	Manejado por un evento	Ninguna	Poner próx. caracter en el buffer	Despreciable

Matrices de Requerimientos (RTM para la C.C.O.)

Requerimientos del Sistema (1*) / Requerimientos del Software (2*)

1* \ 2*	Establecer comunicación con C.C. 1	Procesar datos de cables cortados 2	Generar reportes 3
1.1. Generar reporte de un corte			X
1.2. Generar reporte de una central			X
1.3. Generar reporte completo			X
3.1. Ubica el recorrido del cable en el mapa y busca teléfono		X	
Derivados			
Enviar reconocimiento a C.C	X		
Recibir datos de C.C	X		

Requerimientos del Sistema (1*) / Requerimientos del Software (2*)

1* \ 2*	Recibir paquete 1.1.	Analizar paquete 1.2.	Transmitir a C.C. 1.3.
Enviar reconocimiento a C.C.			X
Recibir datos de C.C.	X		

Requerimientos del Sistema (1*) / Requerimientos del Software (2*)

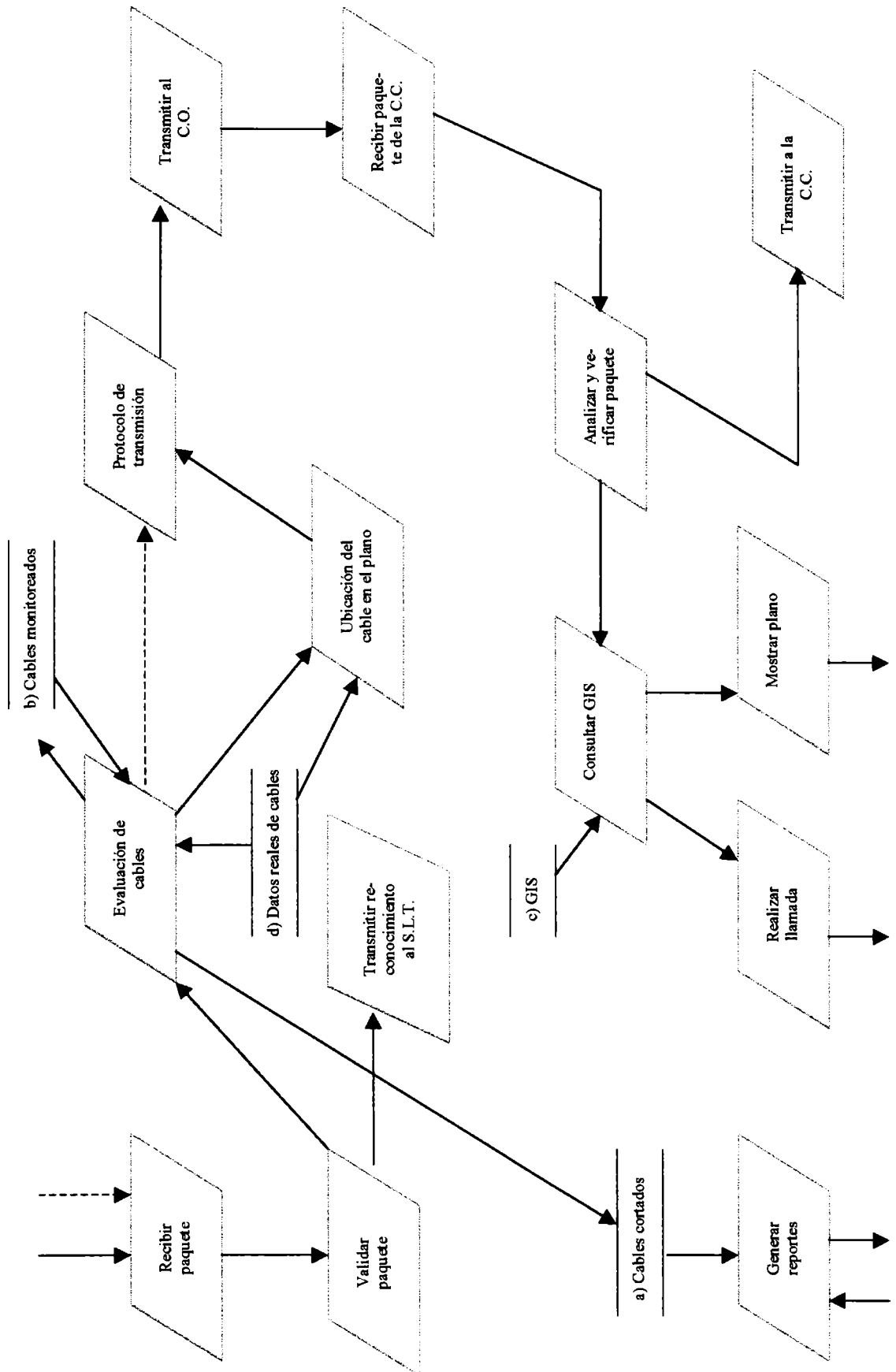
1* \ 2*	Ubicar recorrido del cable en el mapa y teléfono 2.1	Mostrar plano 2.2	Realizar comunicación con la comisaría 2.3
3.1. Ubicar recorrido del cable en el mapa y teléfono	X		
3.2. Mostrar plano		X	
3.3. Realizar comunicación con la comisaría			X

Requerimientos del Sistema (1*) / Requerimientos del Software (2*)

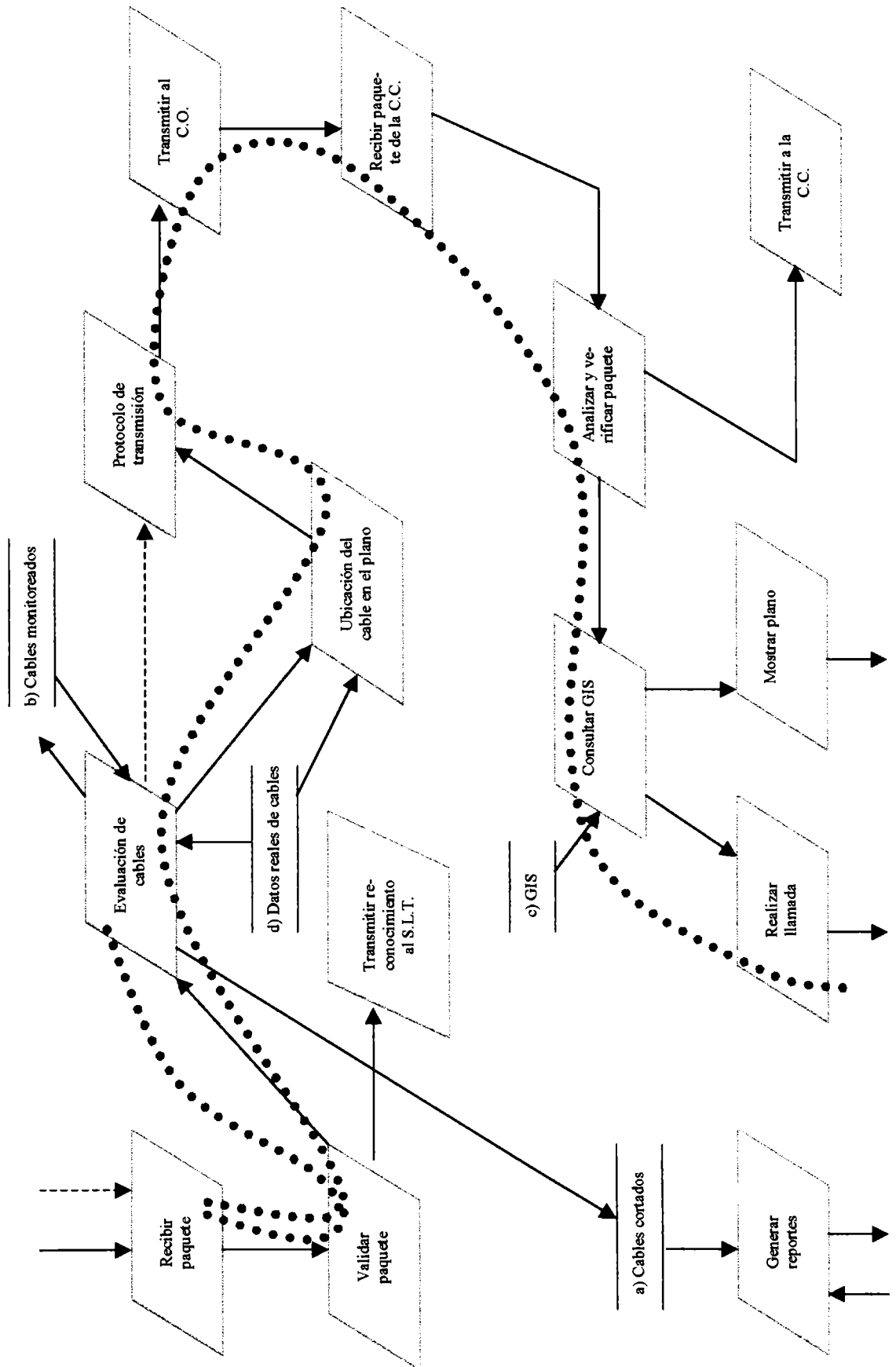
1* \ 2*	Generar reporte de un corte 3.1	Generar reporte de una central 3.2	Generar reporte completo 3.3
1.1. Pedir reporte de un corte	X		
1.2. Pedir reporte de los cortes de una central		X	
1.3. Pedir reporte de todas las centrales			X

Diseño Top Level

Grafo de Comunicación de Tareas (TCG)



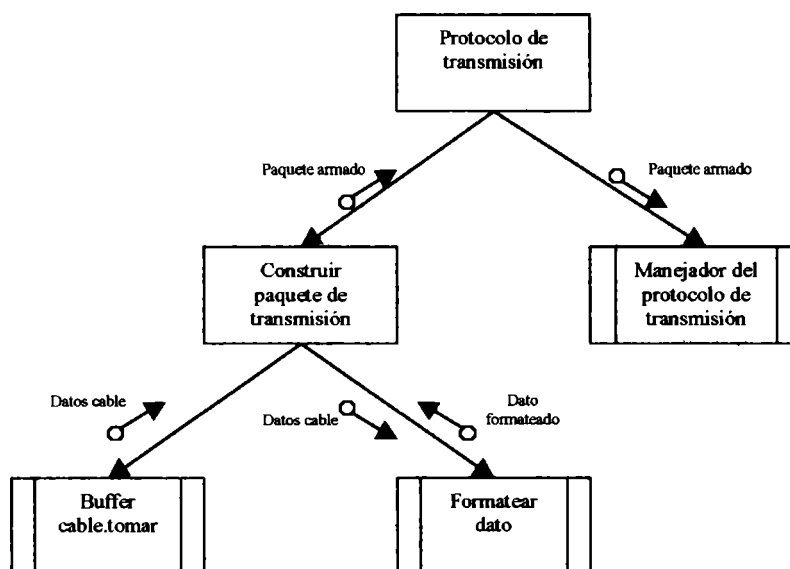
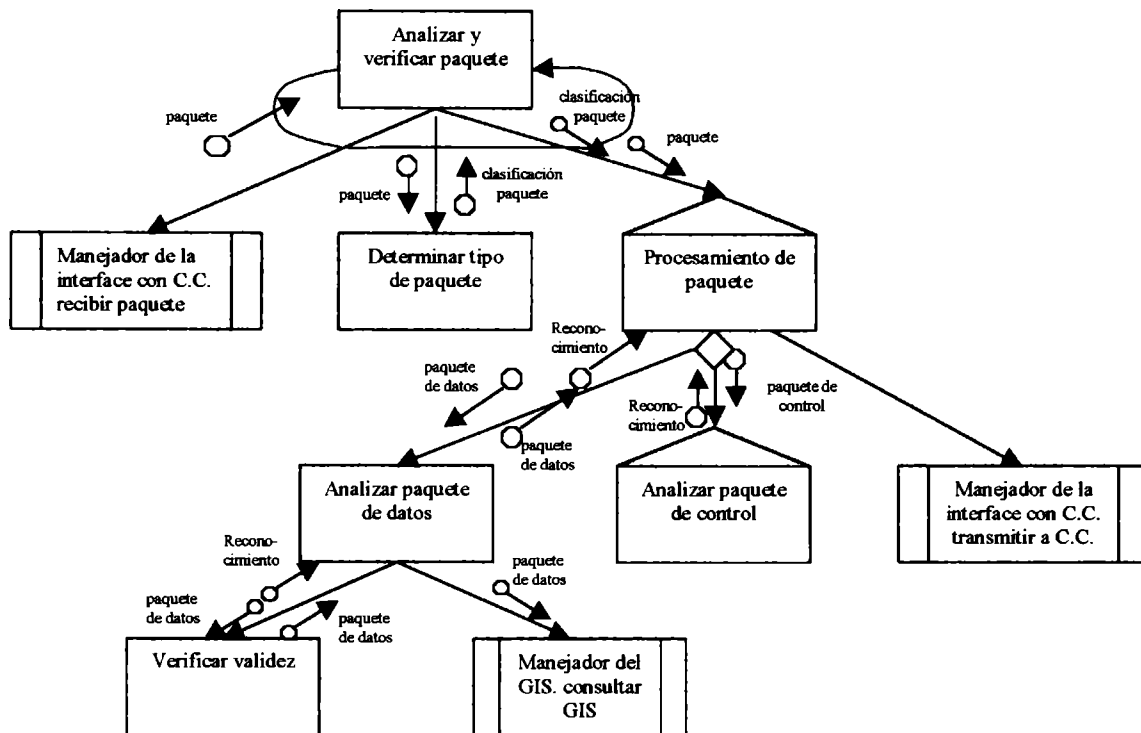
Grafo de Comunicación de Tareas (TCG con tiempo crítico)

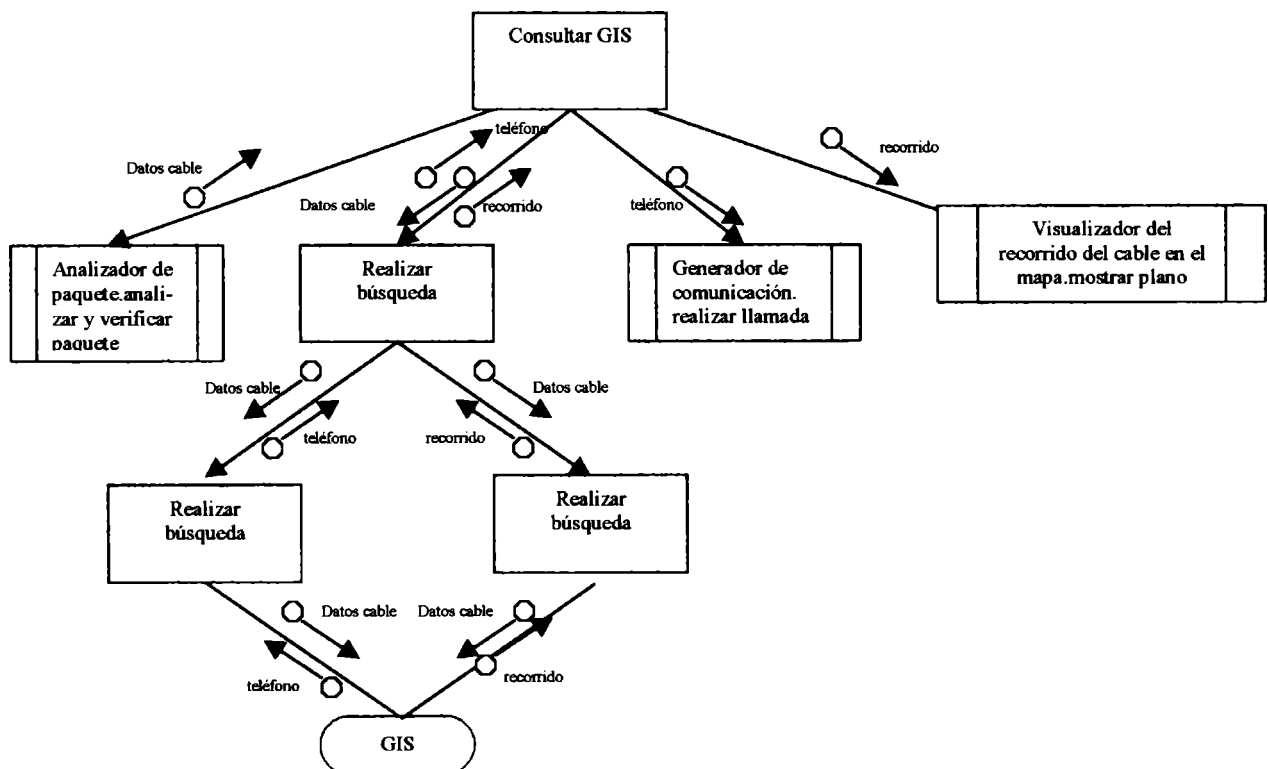
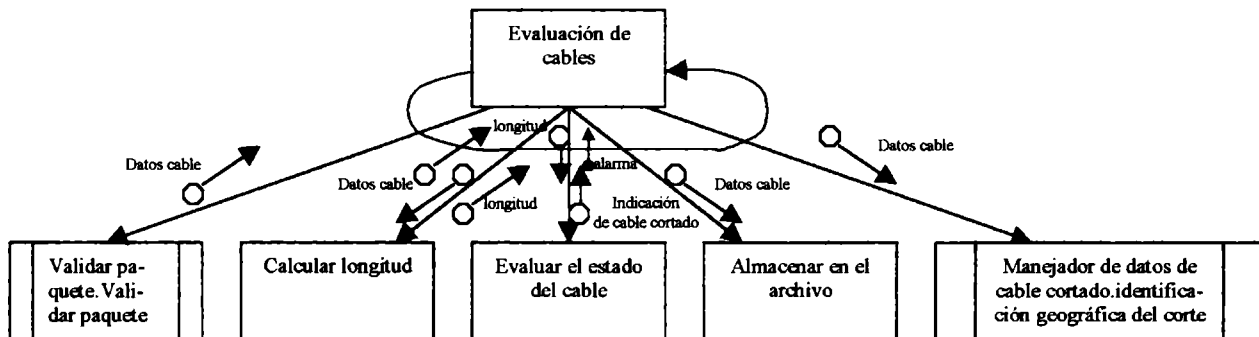


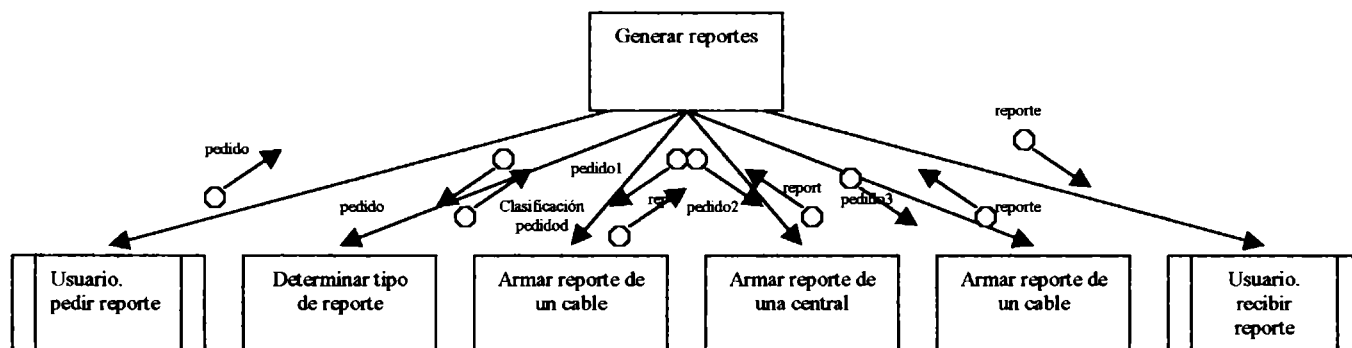
Andrea Artime – Pamela Fernández



Cartas de Estructura







Matriz de requerimientos.

	SLT- Maneja- dor de la interface	Validar paquete	Módulo evaluador de cables	Manejador de datos de cables cortados	Manejador del protocolo de transmisión	Manejador de la interface con la C.C.O.
1. Establecer la comunicación con el S.L.T.	X	X				
1.1. Recibir paquete	X					
1.2. Validar paquete		X				
1.3. Transmitir al S.L.T.	X					
2. Monitorear cables			X	X		
2.1. Obtener longitud real			X			
2.2. Evaluar estado de los cables			X			
2.3. Identificación geográfica del corte				X		
3. Establecer comunicación con la C.C.O.					X	X
3.1. Armar paquete					X	
3.2. Validar paquete					X	
3.3. Manejador protocolo de transmisión					X	
3.4. Transmitir a la C.C.O.						X
1. Establecer comunicación con la C.C.						
1.1. Recibir paquete de C.C.						
1.2. Analizar paquete						
1.2.1. Determinar tipo de paquete						
1.2.2. Validar paquete con señal de control						
1.2.3. Validar paquete con datos						
1.2.4. Recibir validación						
1.3. Transmitir a la C.C.						
2. Procesar datos de los cables cortados						
2.1. Ubicar el recorrido del cable en el mapa y teléfono						
2.2. Mostrar plano						
2.3. Realizar comunicación						
3. Generar reporte						
3.1. Generar reporte completo						
3.2. Generar reporte de una central						
3.3. Generar reporte de un corte						

	Manejador de la interface con la C.C.	Analizador de paquete	Manejador del GIS	Generador de comunicación	Visualizador del recorrido del cable en el mapa	Manejador de reportes	Analizador de paquete
1. Establecer la comunicación con el S.L.T.							
1.1. Recibir paquete							
1.2. Validar paquete							
1.3. Transmitir al S.L.T.							
2. Monitorear cables							
2.1. Obtener longitud real							
2.2. Evaluar estado de los cables							
2.3. Identificación geográfica del corte							
3. Establecer comunicación con la C.C.O.							
3.1. Armar paquete							
3.2. Validar paquete							
3.3. Manejador protocolo de transmisión							
3.4. Transmitir a la C.C.O.							
1. Establecer comunicación con la C.C.							
1.1. Recibir paquete de C.C.	X						
1.2. Analizar paquete							X
1.2.1. Determinar tipo de paquete							X
1.2.2. Validar paquete con señal de control							X
1.2.3. Validar paquete con datos							X
1.2.4. Recibir validación							X
1.3. Transmitir a la C.C.							
2. Procesar datos de los cables cortados			X	X	X		
2.1. Ubicar el recorrido del cable en el mapa y teléfono			X				
2.2. Mostrar plano				X			
2.3. Realizar comunicación					X		
3. Generar reporte						X	
3.1. Generar reporte completo						X	
3.2. Generar reporte de una central						X	
3.3. Generar reporte de un corte						X	

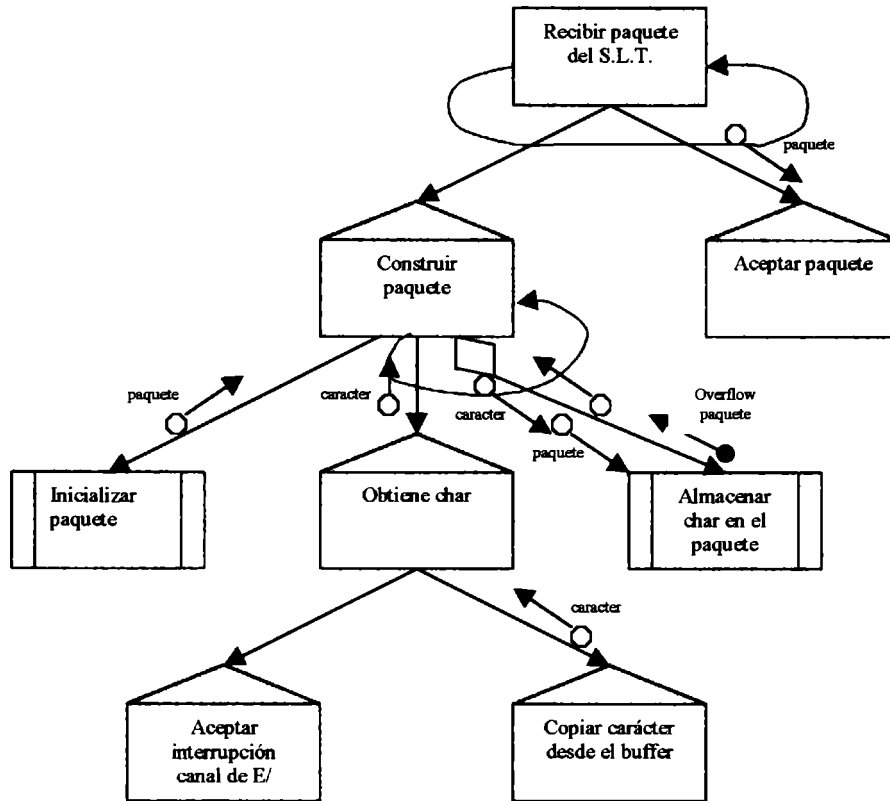
PDL: Lenguaje Pseudo-Código

- **SLT-Manejador de la interface**
Recibe un paquete del S.L.T. y lo envía al *módulo de validación*. Además transmite un reconocimiento al S.L.T.
- **Validar paquete**
Recibe un paquete y le envía un OK al *manejador de la interface de S.L.T.* Además transmite el paquete validado al *módulo evaluador de cables*.
- **Módulo evaluador de cables**
Recibe un paquete validado, evalúa la longitud y determina si está cortado el cable. En caso de ser así, envía los datos al *manejador de cables cortados*. Además transmite los datos del cable al *módulo de registro de cables monitoreados*.
- **Manejador de datos de cables cortados**
Recibe los datos del cable y obtiene la ubicación geográfica del mismo (una identificación del recorrido), y se lo pasa al *manejador de protocolo de transmisión*.
- **Manejador del protocolo de transmisión**
Recibe las coordenadas del cable cortado, arma el paquete según el protocolo de transmisión y lo envía al *módulo manejador de la interface con C.O.*
- **Manejador de la interface con la C.C.O.**
Recibe un paquete armado y lo envía al módulo *manejador de la interface con la C.C.*
- **Registro de cables monitoreados**
Almacena y recupera datos de los cables monitoreados.
- **Registro de cables cortados**
Almacena y recupera datos de los cables cortados.
- **Registro de datos reales de los cables**
Almacena y recupera datos reales de los cables existentes.
- **Manejador de la interface con la C.C.**
Recibe un paquete desde la Central y lo envía al módulo *analizador de paquete*.
- **Analizador de paquete**
Recibe un paquete para analizar y verificar si es de datos o de control y lo envía al módulo *manejador del GIS*.
- **Manejador del GIS**
Consulta el GIS para obtener la ubicación geográfica del cable cortado en el mapa y obtiene los datos del teléfono de la comisaría correspondiente. Envía los datos pertinentes al módulo *generador de comunicación* y al *visualizador del mapa*.

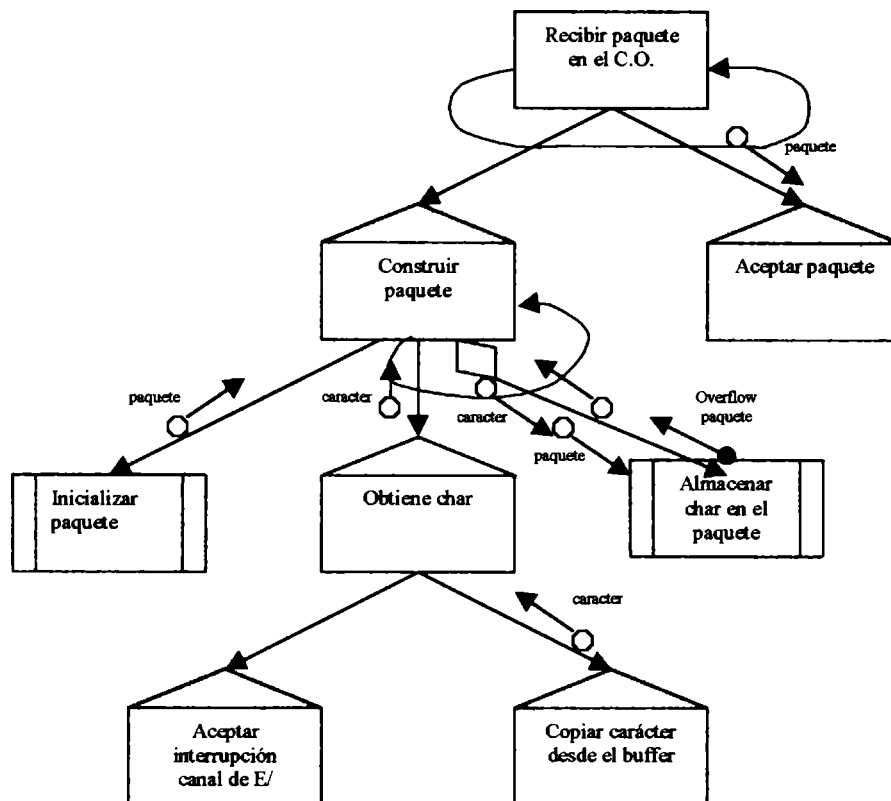
- **Generador de la comunicación**
Recibe los datos del teléfono y realiza la llamada telefónica a la comisaría.
- **Visualizador del recorrido del cable en el mapa**
Recibe los datos del recorrido y lo muestra en el mapa en la C.C.O.
- **Manejador de reportes**
Recibe un pedido de reporte, lo genera y lo envía.
- **Registro del GIS**

Diseño Detallado

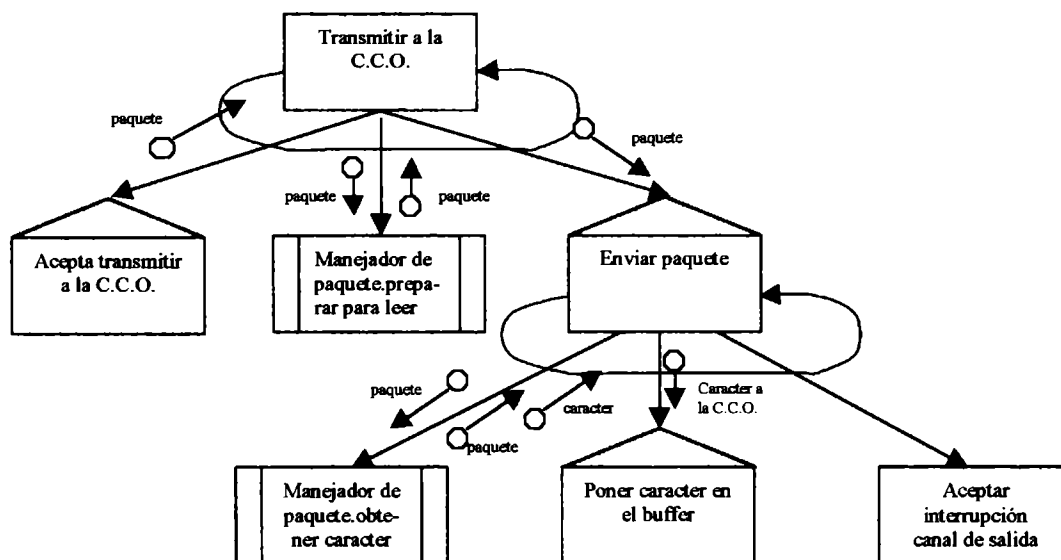
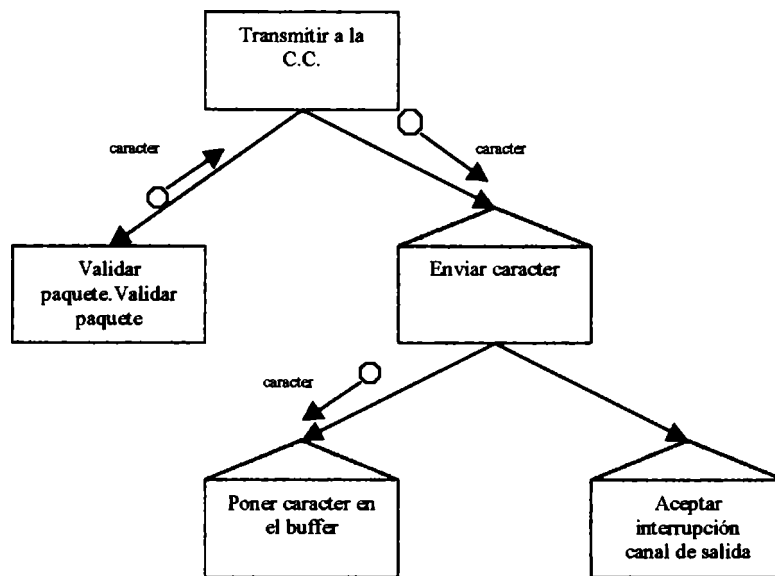
Cartas de Estructura

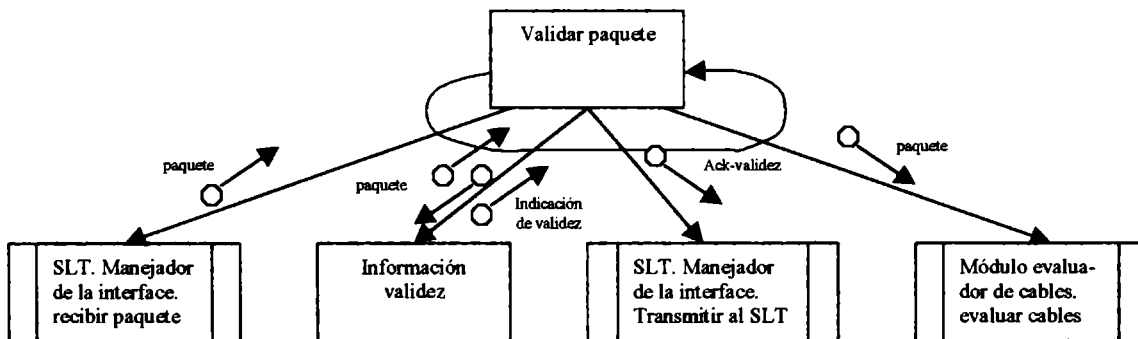
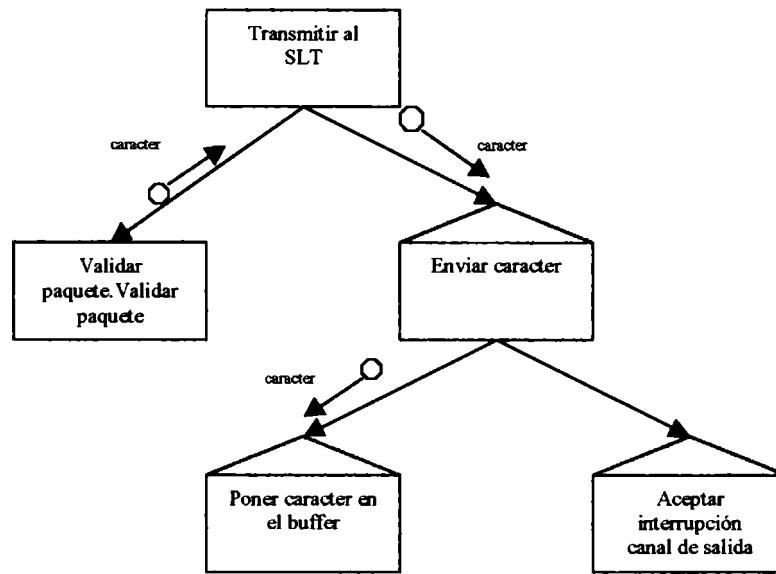


Módulos de ayuda: Manejador de paquete.inicializar
Manejador de paquete.almacenar



Módulos de ayuda: Manejador de paquete.inicializar
Manejador de paquete.almacenar



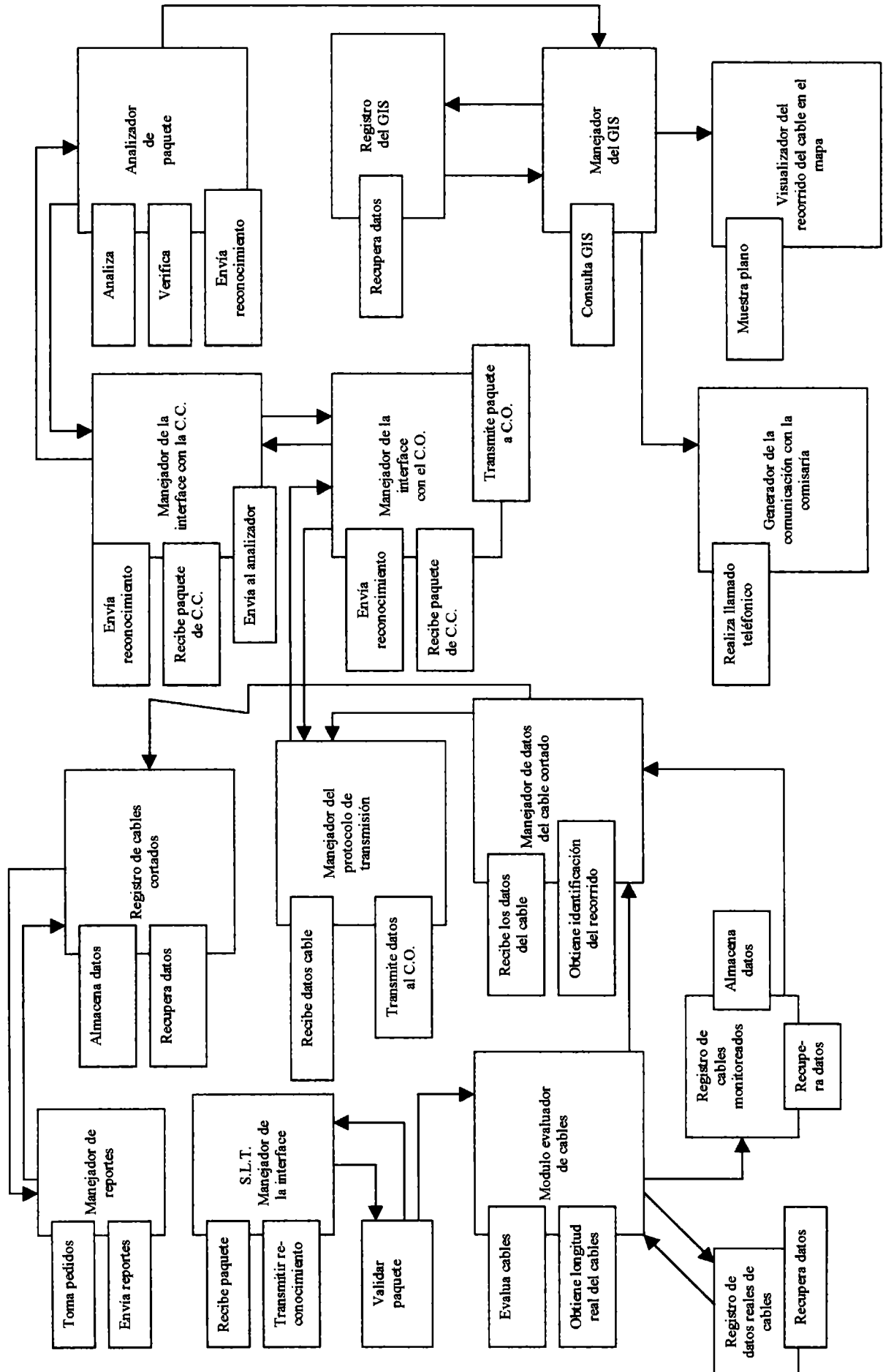


**Identificación
geográfica
del corte**

Mostrar plano

**Realizar
comunicación**

Diagrama de la Arquitectura del Software definitivo (S.A.D.)



PDL: Lenguaje Pseudo-Código

- **SLT-Manejador de la interface**
Recibe un paquete del S.L.T. y lo envía al *módulo de validación*. Además transmite un reconocimiento al S.L.T.
- **Validar paquete**
Recibe un paquete y le envía un OK al *manejador de la interface de S.L.T.* Además transmite el paquete validado al *módulo evaluador de cables*.
- **Módulo evaluador de cables**
Recibe un paquete validado, evalúa la longitud y determina si está cortado el cable. En caso de ser así, envía los datos al *manejador de cables cortados*. Además transmite los datos del cable al *módulo de registro de cables monitoreados*.
- **Manejador de datos de cables cortados**
Recibe los datos del cable y obtiene la ubicación geográfica del mismo (una identificación del recorrido), y se lo pasa al *manejador de protocolo de transmisión*.
- **Manejador del protocolo de transmisión**
Recibe las coordenadas del cable cortado, arma el paquete según el protocolo de transmisión y lo envía al *módulo manejador de la interface con C.O.*
- **Manejador de la interface con la C.C.O.**
Recibe un paquete armado y lo envía al módulo *manejador de la interface con la C.C.*
- **Registro de cables monitoreados**
Almacena y recupera datos de los cables monitoreados.
- **Registro de cables cortados**
Almacena y recupera datos de los cables cortados.
- **Registro de datos reales de los cables**
Almacena y recupera datos reales de los cables existentes.
- **Manejador de la interface con la C.C.**
Recibe un paquete desde la Central y lo envía al módulo *analizador de paquete*.
- **Analizador de paquete**
Recibe un paquete para analizar y verificar si es de datos o de control y lo envía al módulo *manejador del GIS*.
- **Manejador del GIS**
Consulta el GIS para obtener la ubicación geográfica del cable cortado en el mapa y obtiene los datos del teléfono de la comisaría correspondiente. Envía los datos pertinentes al módulo *generador de comunicación* y al *visualizador del mapa*.

- **Generador de la comunicación**
Recibe los datos del teléfono y realiza la llamada telefónica a la comisaría.
- **Visualizador del recorrido del cable en el mapa**
Recibe los datos del recorrido y lo muestra en el mapa en la C.C.O.
- **Manejador de reportes**
Recibe un pedido de reporte, lo genera y lo envía.
- **Registro del GIS**

Matriz de requerimientos.

	SLT- Maneja- dor de la interface	Validar paquete	Módulo evaluador de cables	Manejador de datos de cables cortados	Manejador del protocolo de transmisión	Manejador de la interface con la C.C.O.
1. Establecer la comunicación con el S.L.T.	X	X				
1.1. Recibir paquete	X					
1.2. Validar paquete		X				
1.3. Transmitir al S.L.T.	X					
2. Monitorear cables			X	X		
2.1. Obtener longitud real			X			
2.2. Evaluar estado de los cables			X			
2.3. Identificación geográfica del corte				X		
3. Establecer comunicación con la C.C.O.					X	X
3.1. Armar paquete					X	
3.2. Validar paquete					X	
3.3. Manejador protocolo de transmisión					X	
3.4. Transmitir a la C.C.O.						X
1. Establecer comunicación con la C.C.						
1.1. Recibir paquete de C.C.						
1.2. Analizar paquete						
1.2.1. Determinar tipo de paquete						
1.2.2. Validar paquete con señal de control						
1.2.3. Validar paquete con datos						
1.2.4. Recibir validación						
1.3. Transmitir a la C.C.						
2. Procesar datos de los cables cortados						
2.1. Ubicar el recorrido del cable en el mapa y teléfono						
2.2. Mostrar plano						
2.3. Realizar comunicación						
3. Generar reporte						
3.1. Generar reporte completo						
3.2. Generar reporte de una central						
3.3. Generar reporte de un corte						

	Manejador de la interface con la C.C.	Analizador de paquete	Manejador del GIS	Generador de comunicación	Visualizador del recorrido del cable en el mapa	Manejador de reportes	Analizador de paquete
1. Establecer la comunicación con el S.L.T.							
1.1. Recibir paquete							
1.2. Validar paquete							
1.3. Transmitir al S.L.T.							
2. Monitorear cables							
2.1. Obtener longitud real							
2.2. Evaluar estado de los cables							
2.3. Identificación geográfica del corte							
3. Establecer comunicación con la C.C.O.							
3.1. Armar paquete							
3.2. Validar paquete							
3.3. Manejador protocolo de transmisión							
3.4. Transmitir a la C.C.O.							
1. Establecer comunicación con la C.C.							
1.1. Recibir paquete de C.C.	X						
1.2. Analizar paquete							X
1.2.1. Determinar tipo de paquete							X
1.2.2. Validar paquete con señal de control							X
1.2.3. Validar paquete con datos							X
1.2.4. Recibir validación							X
1.3. Transmitir a la C.C.							
2. Procesar datos de los cables cortados			X	X	X		
2.1. Ubicar el recorrido del cable en el mapa y teléfono			X				
2.2. Mostrar plano				X			
2.3. Realizar comunicación					X		
3. Generar reporte						X	
3.1. Generar reporte completo						X	
3.2. Generar reporte de una central						X	
3.3. Generar reporte de un corte						X	

2. Análisis de Tiempos

Módulos a tener en cuenta para el análisis de los tiempos:

1. **Recibir la información de SLT.**
2. **Determinar si está cortado el cable leído.**
3. **Establecer la comunicación para transmitir los datos.**
4. **Transmisión de los datos.**
5. **Buscar datos en el GIS.**
6. **Establecer la comunicación con la comisaría.**
7. **Transmitirle los datos.**

1. **Recibir la información de SLT.**

El monitoreo de los cables se realiza cada 1 segundo. El análisis de nuestros tiempos comenzará una vez que tenemos la lectura de la longitud del cable monitoreado.

2. **Determinar si está cortado el cable leído.**

Una vez leída la longitud del cable se accede a un archivo para obtener la longitud real del cable. Luego se compara y se determina si está o no cortado el cable.

Este módulo se podrá realizar en **13 mseg.** aproximadamente. Este dato fue calculado tomando una muestra de 50 cables cortados y en una máquina (1).

3. **Establecer la comunicación para transmitir los datos.**

Tomando como base un módem de 33600 bps, un par telefónico como medio de comunicación y una comunicación máquina-máquina, el tiempo para establecer la comunicación está determinado por el número de rings que se ejecutan hasta obtener respuesta (1 seg. por ring. aproximadamente), y el tiempo de negociación entre los dos módems, dependiendo de la velocidad de ambos módems. Al ser una comunicación entre dos máquinas, podemos considerar que la cantidad de rings estipulado es de uno. De acuerdo a las pruebas realizadas, se puede concluir que el tiempo para establecer la comunicación es aproximadamente de **10 segundos**. Este tiempo se incrementará en el caso de que la línea no esté disponible y haya que reintentar la comunicación. El incremento del tiempo será de 1 seg. aproximadamente por cada reintento.

4. **Transmisión de los datos.**

De acuerdo a la velocidad del módem mencionado anteriormente, y al tamaño de la información a ser transferida, el tiempo aproximado para la transmisión de los datos es menor a **1 seg.**

5. **Buscar datos referentes a la comisaría y teléfono, y mostrar el plano.**

Este módulo comienza cuando se recibe la información de un cable cortado, el cual debe ubicarse geográficamente para identificar la comisaría correspondiente a la zona del corte. El tiempo va a depender del tamaño del mapa y la cantidad de recorridos trazados. Aproximado sería de 10 seg.

6. **Establecer la comunicación con la comisaría.**

Tomando como base un módem de 33600 bps, un par telefónico como medio de comunicación, los tipos de comunicación que se pueden establecer son: máquina-máquina,

máquina-fax, máquina-hombre. En el primer caso el tiempo estimado es igual al del punto 3, teniendo en cuenta la mismas consideraciones y los mismos retrasos. En el segundo caso la cantidad de rings puede incrementarse lo que determina un aumento en el tiempo. En el tercer caso, al ser atendido el teléfono por una persona aumenta considerablemente el tiempo, ya que la cantidad mínima de rings debería ser de 3.

7. Transmisión de los datos.

De acuerdo a la velocidad del módem mencionado anteriormente, y al tamaño de la información a ser transferida, el tiempo dependerá del tipo de comunicación establecida. En el caso de una comunicación máquina-máquina y el tiempo será aproximadamente igual al punto 4. En el caso de una comunicación máquina-fax el tiempo será aproximadamente de ya que la información a enviar es En el caso de una comunicación máquina-hombre el tiempo se incrementará ya que el módem debe detectar que atendió una persona y enviar un mensaje hablado, cuyo volumen de información es mayor que en el caso anterior. Además debe ser un módem con reconocimiento de voz.

El mejor de los casos sería aquel en el que detectamos un corte y automáticamente los transmitimos al centro de operaciones sin encontrar la línea de comunicación ocupada. El centro de operaciones realiza la ubicación geográfica y se comunica con la comisaría correspondiente a la zona del corte encontrando la línea disponible. El tiempo total aproximado sería de 35 segundos si la comunicación entre el Centro de Operaciones y la comisaría es máquina-máquina, y de 65 segundos aproximadamente si la comunicación entre el Centro de Operaciones y la comisaría es máquina-hombre.

Los inconvenientes con los que nos podríamos encontrar serían los siguientes:

- a) Al intentar la central comunicarse con el centro de operaciones, éste esté atendiendo otro llamado y la línea se encuentre ocupada.
 - b) El centro de operaciones recibió los datos de un cable cortado pero está buscando la información geográfica de un corte anterior, con lo cual se retrasa la búsqueda de los datos de este corte.
 - c) El centro de operaciones recibió los datos de un cable cortado y ya buscó la información geográfica del corte, pero el módem está ocupado con la transmisión anterior.
 - d) El cable telefónico que une la central y el centro de operaciones está cortado con lo que los cortes no pueden ser reportados.
 - e) El cable telefónico que une el centro de operaciones y la comisaría está cortado con lo que los cortes no pueden ser reportados.
-
- Considerando que la línea telefónica entre la central y el centro de operaciones esté ocupada y la línea telefónica que une el centro de operaciones con la comisaría esté libre y sea máquina-máquina, el número máximo de intentos de comunicación desde la central al centro de operaciones para poder cumplir con el tiempo máximo estipulado debería ser de 145 intentos, considerando cada reintento de 1 segundo.
 - Considerando que la línea telefónica entre la central y el centro de operaciones esté libre y la línea telefónica que une el centro de operaciones con la comisaría esté ocupada el número máximo de intentos de comunicación desde el centro de operaciones y la comisaría para poder cumplir con el tiempo máximo estipulado debería ser de 145 intentos si la comunicación es máquina-máquina, de xx intentos si la comunicación es máquina-fax y de 115 intentos si la comunicación es máquina-hombre.

- Considerando que la línea telefónica entre la central y el centro de operaciones esté ocupada y la línea telefónica que une el centro de operaciones con la comisaría esté ocupada también, la sumatoria del número máximo de intentos de comunicación desde la central al centro de operaciones y desde el centro de operaciones a la comisaría debería ser de 145 intentos si la comunicación es maquina-máquina, de xx intentos si la comunicación es maquina-fax y de 115 intentos si la comunicación es maquina-hombre.

Posibles soluciones a los inconvenientes encontrados:

- Para el caso en que la línea telefónica que une alguna central con el centro de operaciones esté cortada, se podría tener en el centro de operaciones un proceso que esté monitoreando la o las líneas telefónicas que lo unen con las centrales.
- Para el caso en que la línea telefónica que une el centro de operaciones con la comisaría correspondiente esté ocupada, se podría tener un número telefónico de alguna comisaría alternativa a la cual llamar luego de xx intentos de comunicación fallidos, igualmente si no puede establecerse la comunicación por corte de línea telefónica.
- En el caso de no poder buscar los datos geográficos de un corte por estar buscando los datos de otro, se podrían tener threads de búsqueda, para lo cual necesitaríamos un sistema operativo multitarea adecuado para esta solución, con lo que tendríamos varias instancias de búsqueda y se aceleraría el proceso.

Para poder verificar la cantidad de cables que se pueden atender simultáneamente, monitoreados cada 1 segundo, confeccionamos 42 Redes de Petri, teniendo en cuenta cantidad de cables (1, 4 y 10), si están todos cortados o no, si hay reintentos en alguna o ambas de las comunicaciones, el tipo de comunicación entre el centro de operaciones y la comisaría y los tiempos mencionados anteriormente. Posteriormente realizamos 20 ejecuciones de cada una y calculamos el tiempo promedio de cada alternativa.

1. 1 cable sin reintentos, comunicación comisaría (máq-máq), con posibilidad de que no este cortado
Tiempo promedio: 35.05 seg.
2. 4 cables sin reintentos, comunicación comisaría (máq-máq), con posibilidad de que no este cortado.
Tiempo promedio: 42.95 seg.
3. 10 cables sin reintentos, comunicación comisaría (máq-máq), con posibilidad de que no este cortado .
Tiempo promedio: 49.65 seg.
4. 1 cable sin reintentos, comunicación comisaría (máq-máq), todos cortados.
Tiempo promedio: 37.27 seg.
5. 4 cables sin reintentos, comunicación comisaría (máq-máq), todos cortados .
Tiempo promedio: 45.54 seg.

6. 10 cables sin reintentos, comunicación comisaría (máq-máq), todos cortados Tiempo promedio: 50.28 seg.
7. 1 cable sin reintentos, comunicación comisaría (máq-hom), con posibilidad de que no este cortado .
Tiempo promedio: 65.24 seg.
8. 4 cables sin reintentos, comunicación comisaría (máq-hom), con posibilidad de que no este cortado .
Tiempo promedio: 78.43 seg.
9. 10 cables sin reintentos, comunicación comisaría (máq-hom), con posibilidad de que no este cortado.
Tiempo promedio: 85.16 seg.
10. 1 cable sin reintentos, comunicación comisaría (máq-hom), todos cortados .
Tiempo promedio: 67.47 seg.
11. 4 cables sin reintentos, comunicación comisaría (máq-hom), todos cortados .
Tiempo promedio: 78.13 seg.
12. 10 cables sin reintentos, comunicación comisaría (máq-hom), todos cortados Tiempo promedio: 85.13 seg.
13. 1 cable con reintentos comunicación CC-CO, con posibilidad de que no este cortado.
Tiempo promedio: 40.18 seg.
14. 4 cables con reintentos comunicación CC-CO, con posibilidad de que no este cortado.
Tiempo promedio: 51.68 seg.
15. 10 cables con reintentos comunicación CC-CO, con posibilidad de que no este cortado.
Tiempo promedio: 63.06 seg.
16. 1 cable con reintentos comunicación CC-CO, todos cortados.
Tiempo promedio: 43.54 seg.
17. 4 cables con reintentos comunicación CC-CO, todos cortados.
Tiempo promedio: 17.49 seg.
18. 10 cables con reintentos comunicación CC-CO, todos cortados.
Tiempo promedio: 68.54 seg.
19. 1 cable con reintentos comunicación CO-comisaría (máq-máq), con posibilidad de que no este cortado.
Tiempo promedio: 44.81 seg.
20. 4 cables con reintentos comunicación CO-comisaría (máq-máq), con posibilidad de que no este cortado.
Tiempo promedio: 48.98 seg.

21. 10 cables con reintentos comunicación CO-comisaría (máq-máq), con posibilidad de que no este cortado.
Tiempo promedio: 59.94 seg.
22. 1 cable con reintentos comunicación CO-comisaría (máq-máq), todos cortados. Tiempo promedio: 38.46 seg.
23. 4 cables con reintentos comunicación CO-comisaría (máq-máq), todos cortados. Tiempo promedio: 53.60 seg.
24. 10 cables con reintentos comunicación CO-comisaría (máq-máq), todos cortados. Tiempo promedio: 62.90 seg.
25. 1 cable con reintentos comunicación CO-comisaría (máq-hom), con posibilidad de que no este cortado.
Tiempo promedio: 71.32 seg.
26. 4 cables con reintentos comunicación CO-comisaría (máq-hom), con posibilidad de que no este cortado.
Tiempo promedio: 154.98 seg.
27. 10 cables con reintentos comunicación CO-comisaría (máq-hom), con posibilidad de que no este cortado.
Tiempo promedio: 126.36 seg.
28. 1 cable con reintentos comunicación CO-comisaría (máq-hom), todos cortados. Tiempo promedio: 83.69 seg.
29. 4 cables con reintentos comunicación CO-comisaría (máq-hom), todos cortados. Tiempo promedio: 102.91 seg.
30. 10 cables con reintentos comunicación CO-comisaría (máq-hom), todos cortados. Tiempo promedio: 125.31 seg.
31. 1 cable con reintentos comunicación CC-CO / CO-comisaría (máq-máq), con posibilidad de que no este cortado.
Tiempo promedio: 46.46 seg.
32. 4 cables con reintentos comunicación CC-CO / CO-comisaría (máq-máq), con posibilidad de que no este cortado.
Tiempo promedio: 58.23 seg.
33. 10 cables con reintentos comunicación CC-CO / CO-comisaría (máq-máq), con posibilidad de que no este cortado.
Tiempo promedio: 71.15 seg.
34. 1 cable con reintentos comunicación CC-CO / CO-comisaría (máq-máq), todos cortados.
Tiempo promedio: 46.57 seg.
35. 4 cables con reintentos comunicación CC-CO / CO-comisaría (máq-máq), todos cortados.

Tiempo promedio: 65.54 seg.

36. 10 cables con reintentos comunicación CC-CO / CO-comisaría (máq-máq), todos cortados.

Tiempo promedio: 71.45 seg.

37. 1 cable con reintentos comunicación CC-CO / CO-comisaría (máq-hom), con posibilidad de que no este cortado.

Tiempo promedio: 90.43 seg.

38. 4 cables con reintentos comunicación CC-CO / CO-comisaría (máq-hom), con posibilidad de que no este cortado.

Tiempo promedio: 106.92 seg.

39. 10 cables con reintentos comunicación CC-CO / CO-comisaría (máq-hom), con posibilidad de que no este cortado.

Tiempo promedio: 142.58 seg.

40. 1 cable con reintentos comunicación CC-CO / CO-comisaría (máq-hom), todos cortados.

Tiempo promedio: 78.98 seg.

41. 4 cables con reintentos comunicación CC-CO / CO-comisaría (máq-hom), todos cortados.

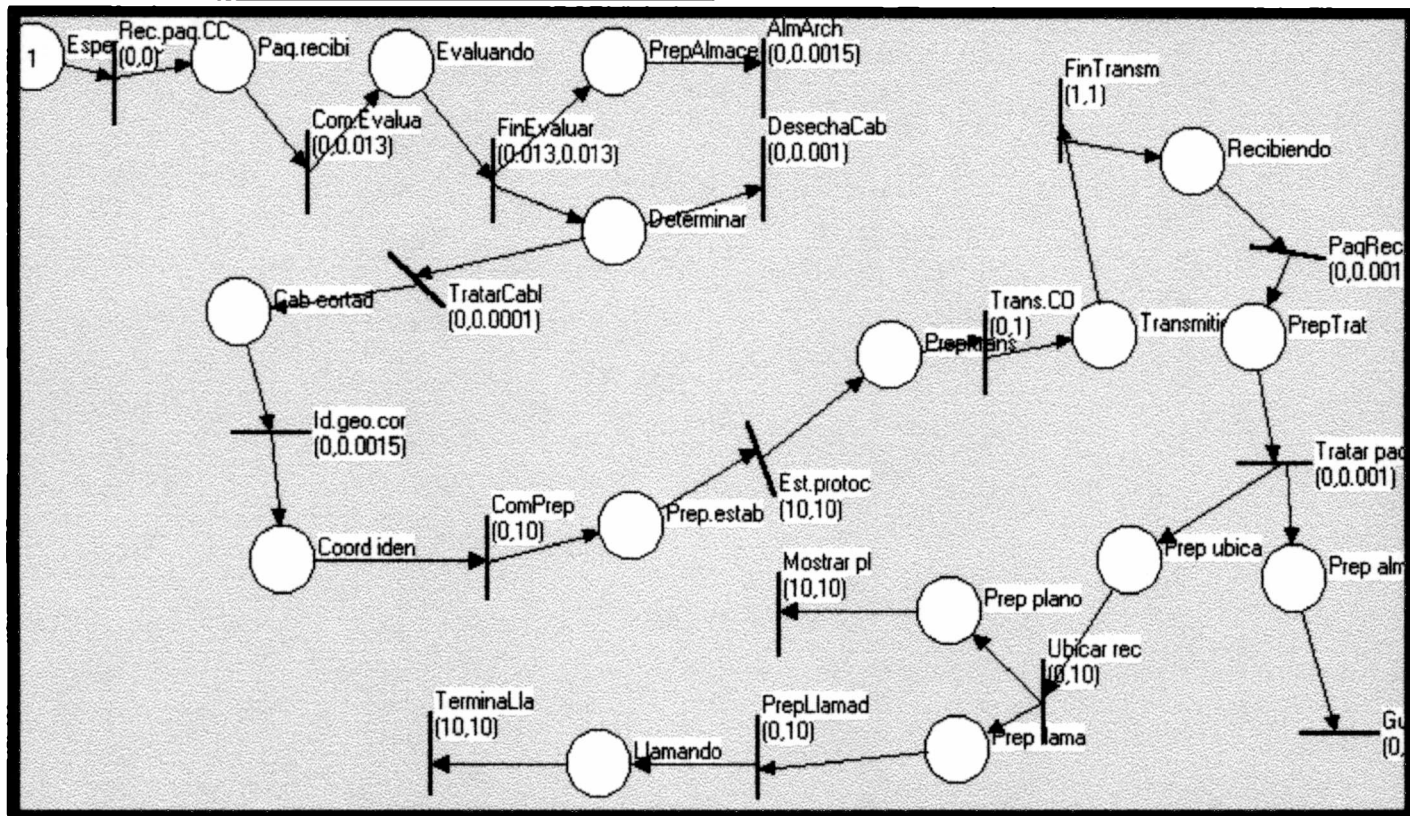
Tiempo promedio: 113.52 seg.

42. 10 cables con reintentos comunicación CC-CO / CO-comisaría (máq-hom), todos cortados.

Tiempo promedio: 130.59 seg.

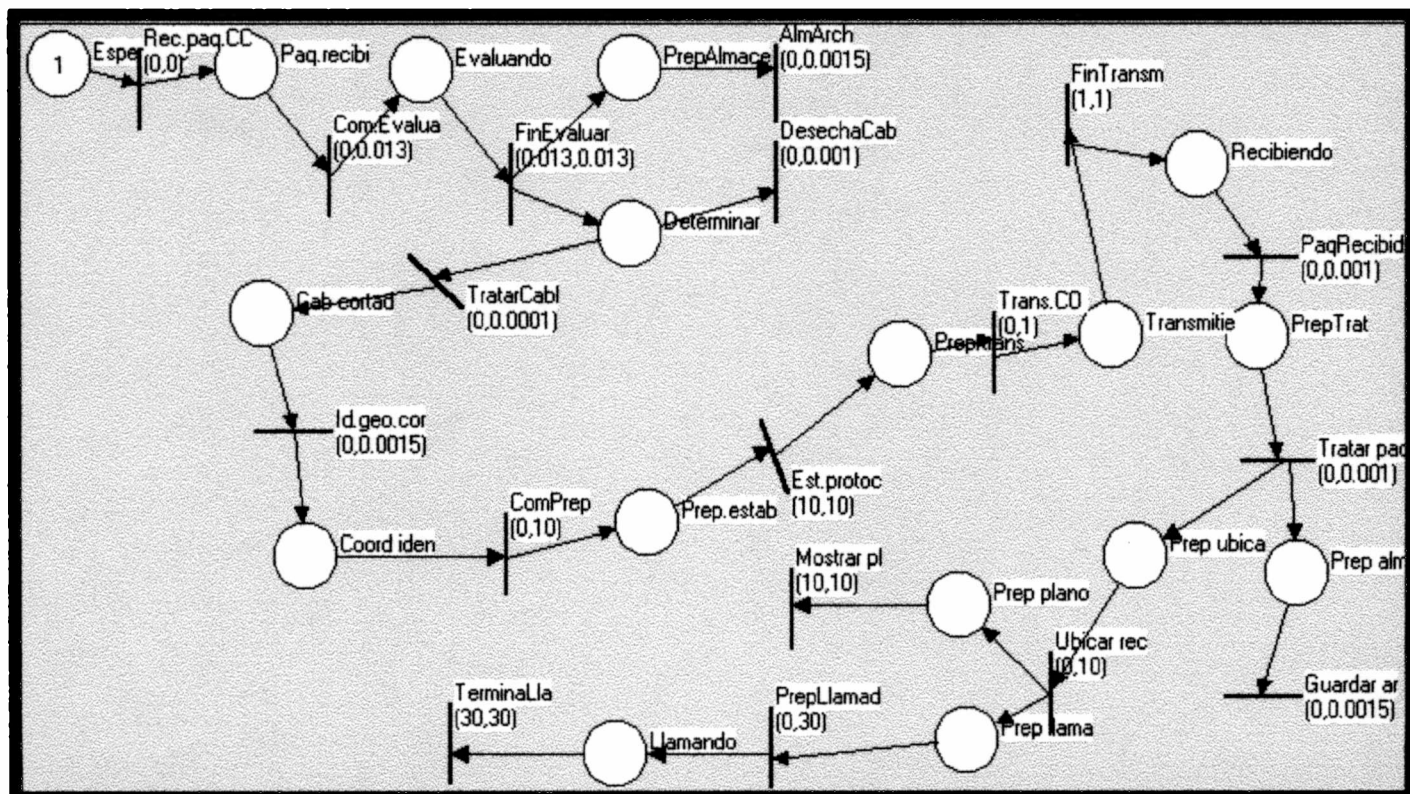
En las alternativas 27, 41 y 42, analizando las 20 ejecuciones observamos que en ciertas circunstancias en 3 minutos no podía realizar la comunicación con la comisaría para algunos cables ya que el número de reintentos superaba el tiempo máximo estipulado.

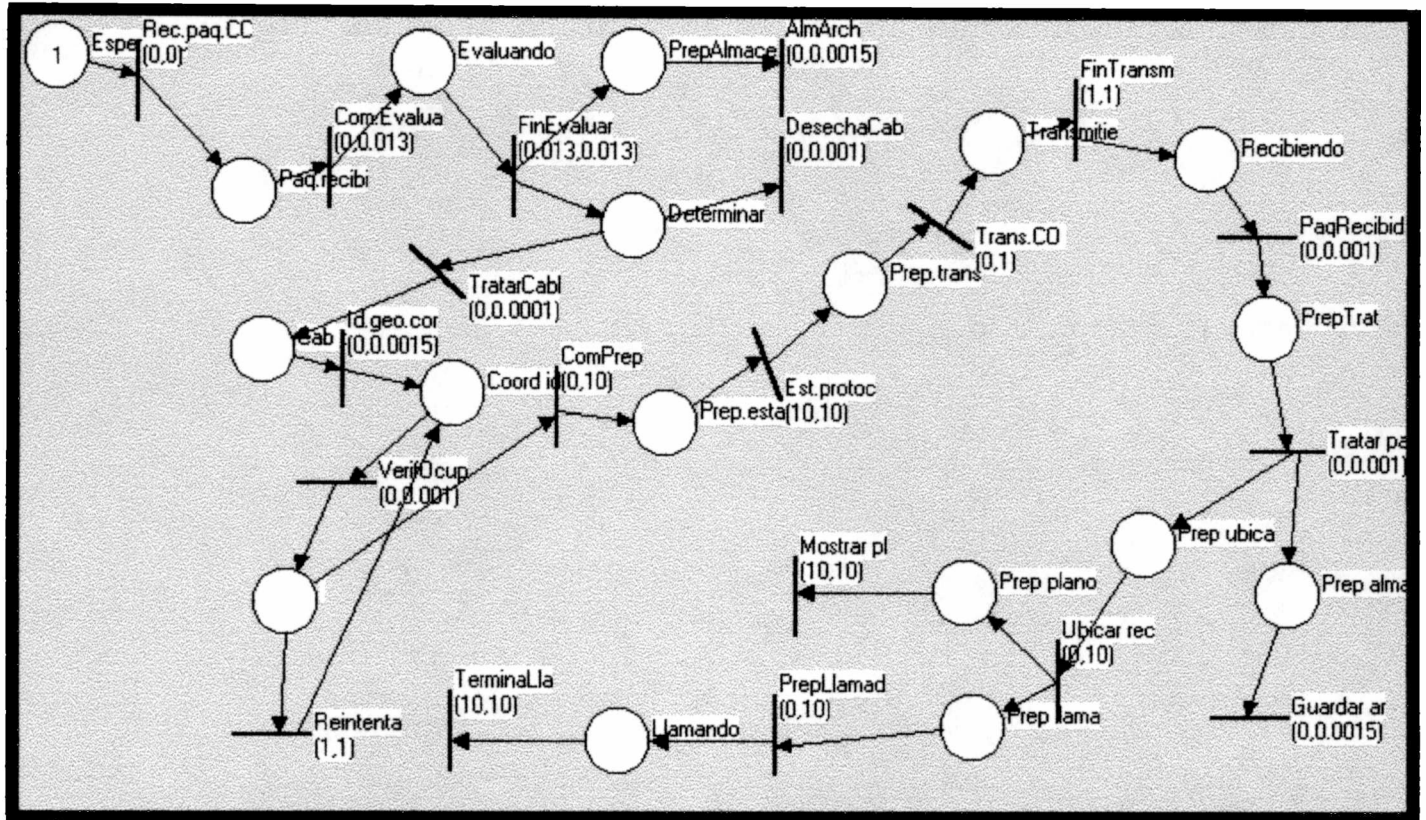
3. Algunas Redes de Petri implementadas



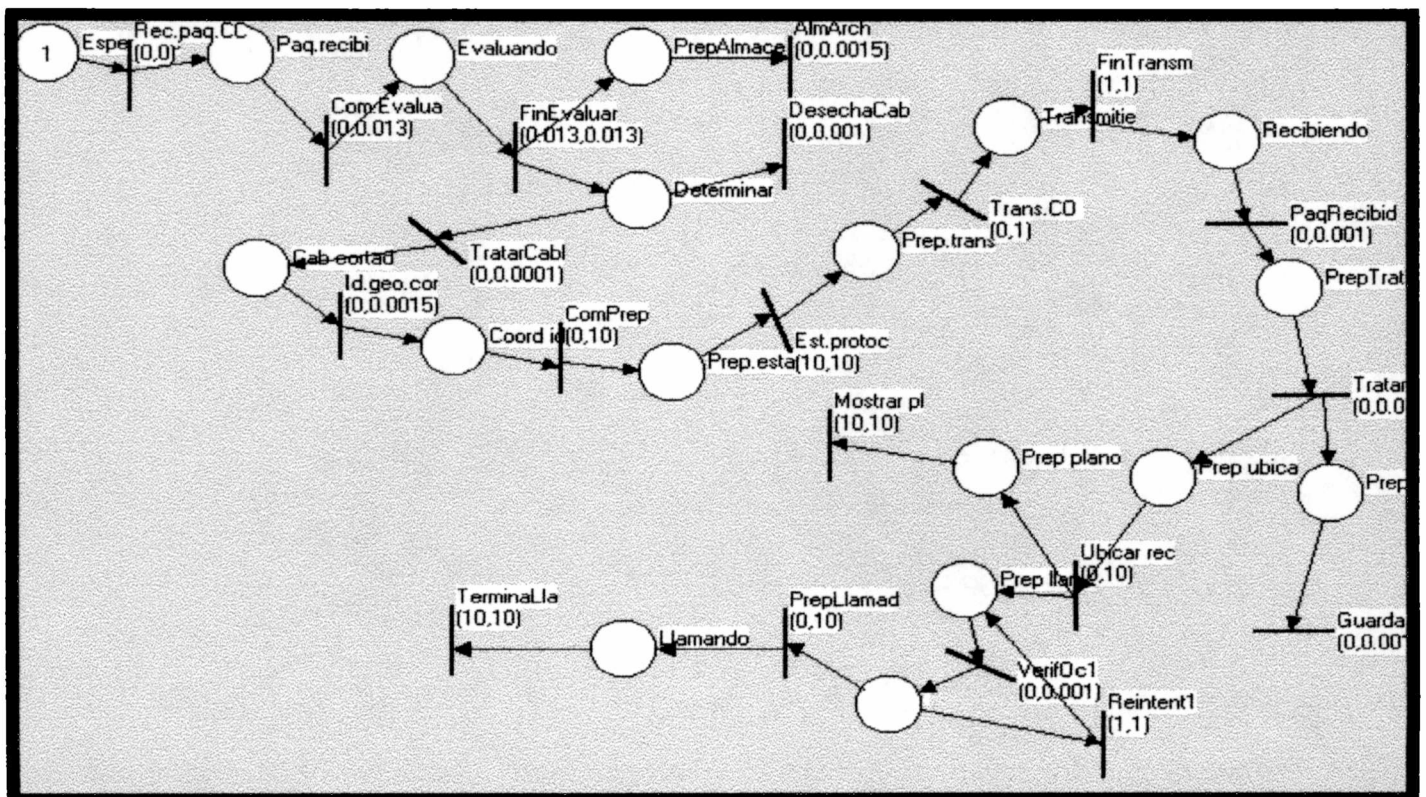
Arriba: Red de Petri con comunicaciones sin reintentos, y máquina-máquina.

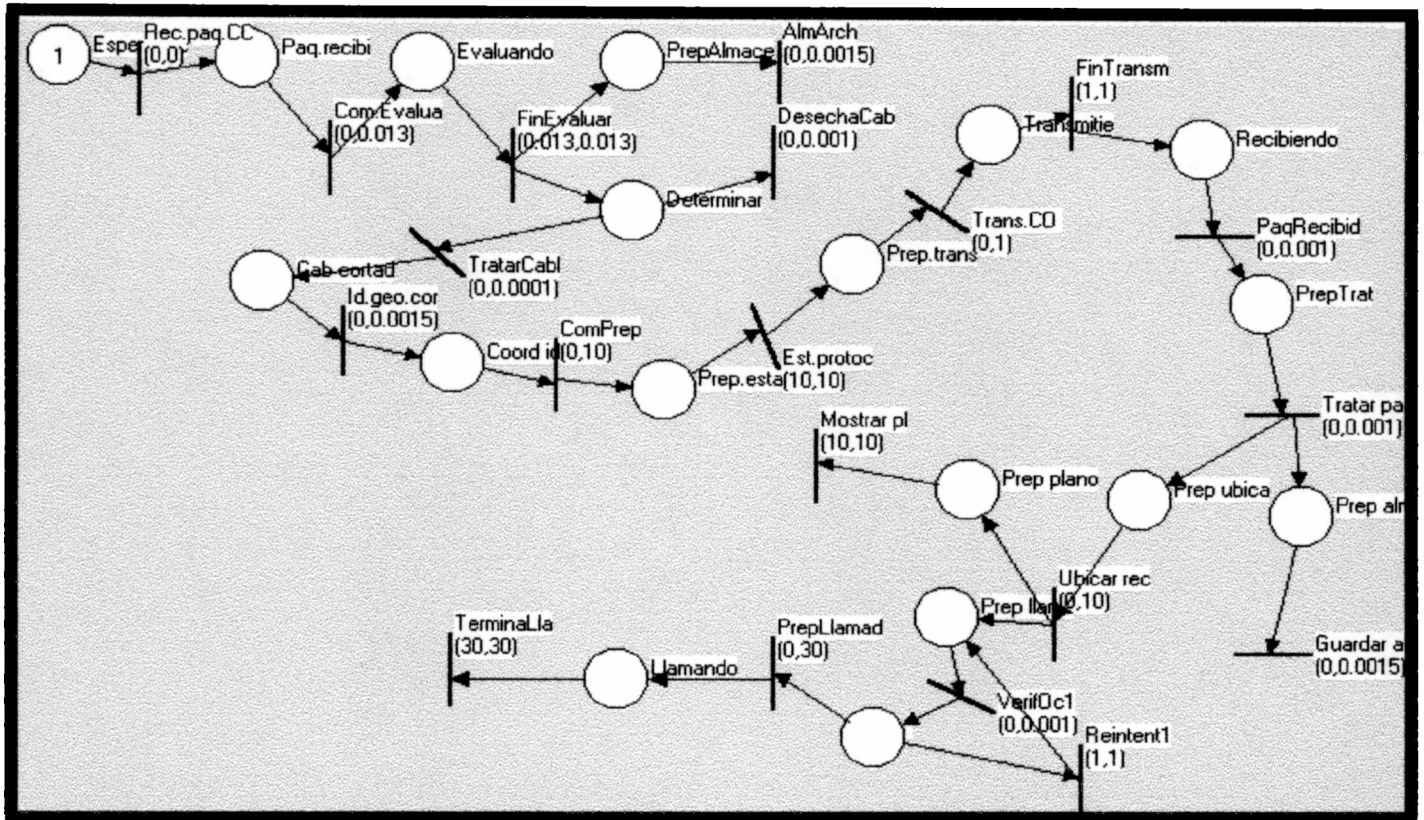
Abajo: Red de Petri con comunicaciones sin reintentos, con comunicación máquina-máquina-Hombre entre el Centro de Operaciones y la Comisaría.





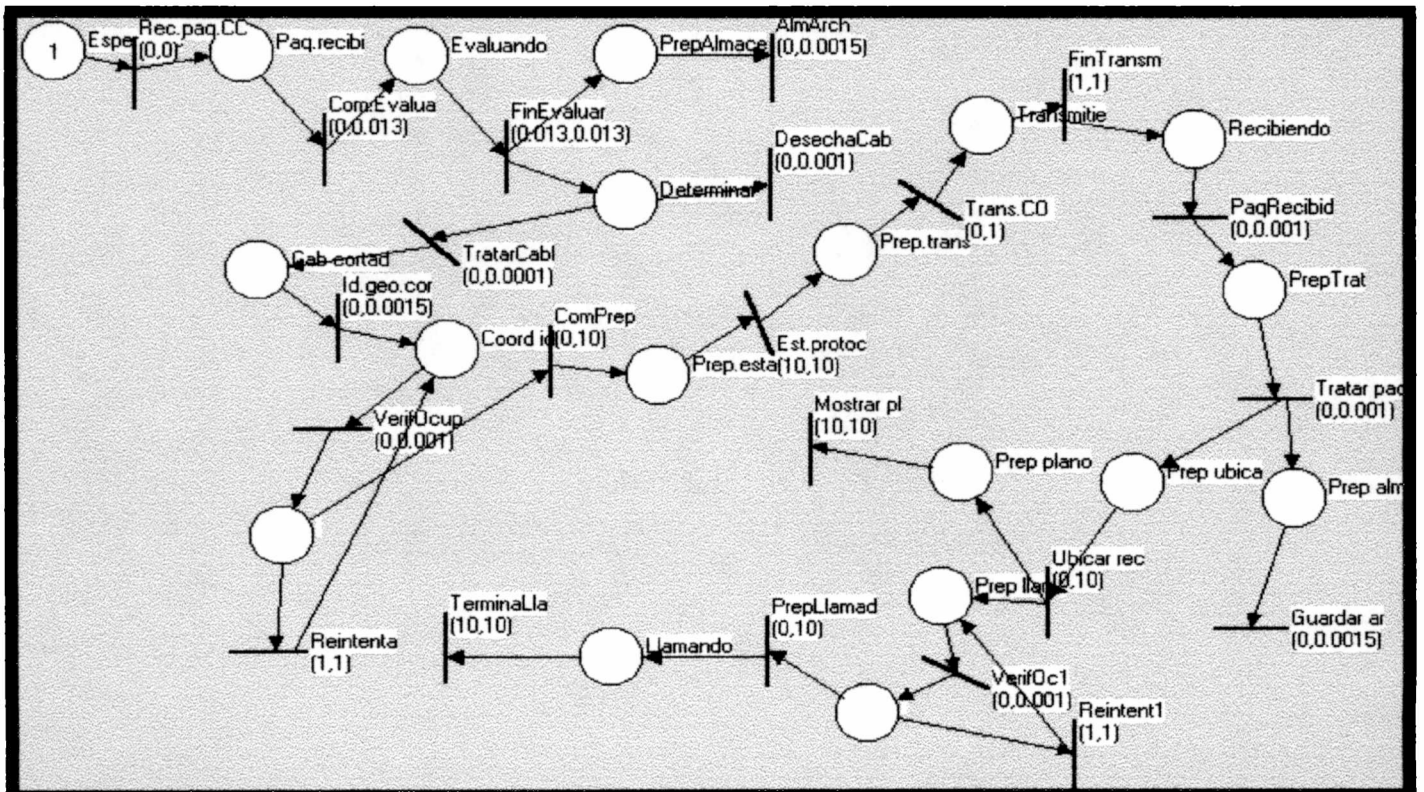
Arriba: Red de Petri con reintentos en la comunicación entre la Central y el Centro de Operaciones, comunicaciones máquina-máquina.
 Abajo: Red de Petri con reintentos en la comunicación entre el Centro de Operaciones y la Comisaria, con comunicaciones máquina-máquina.

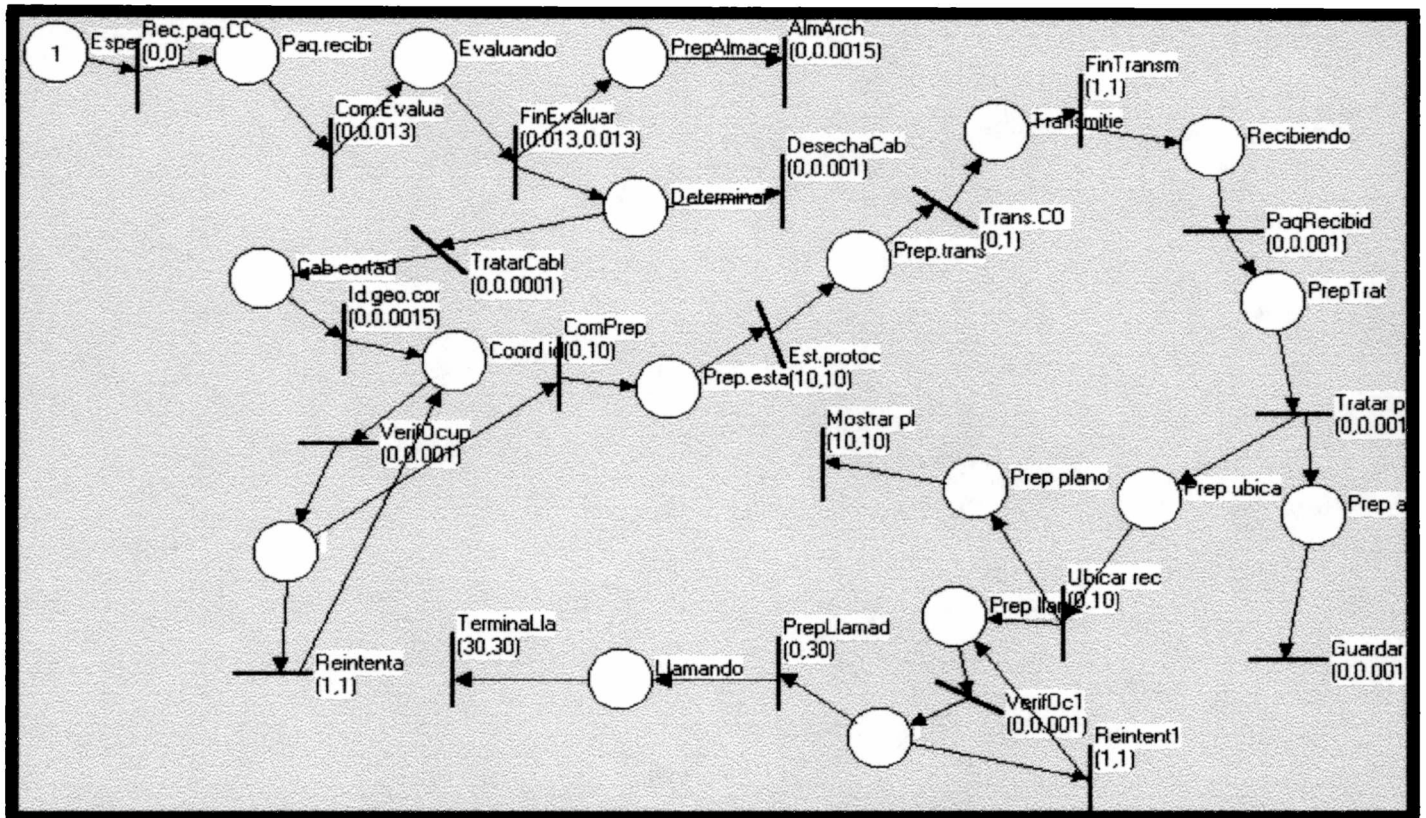




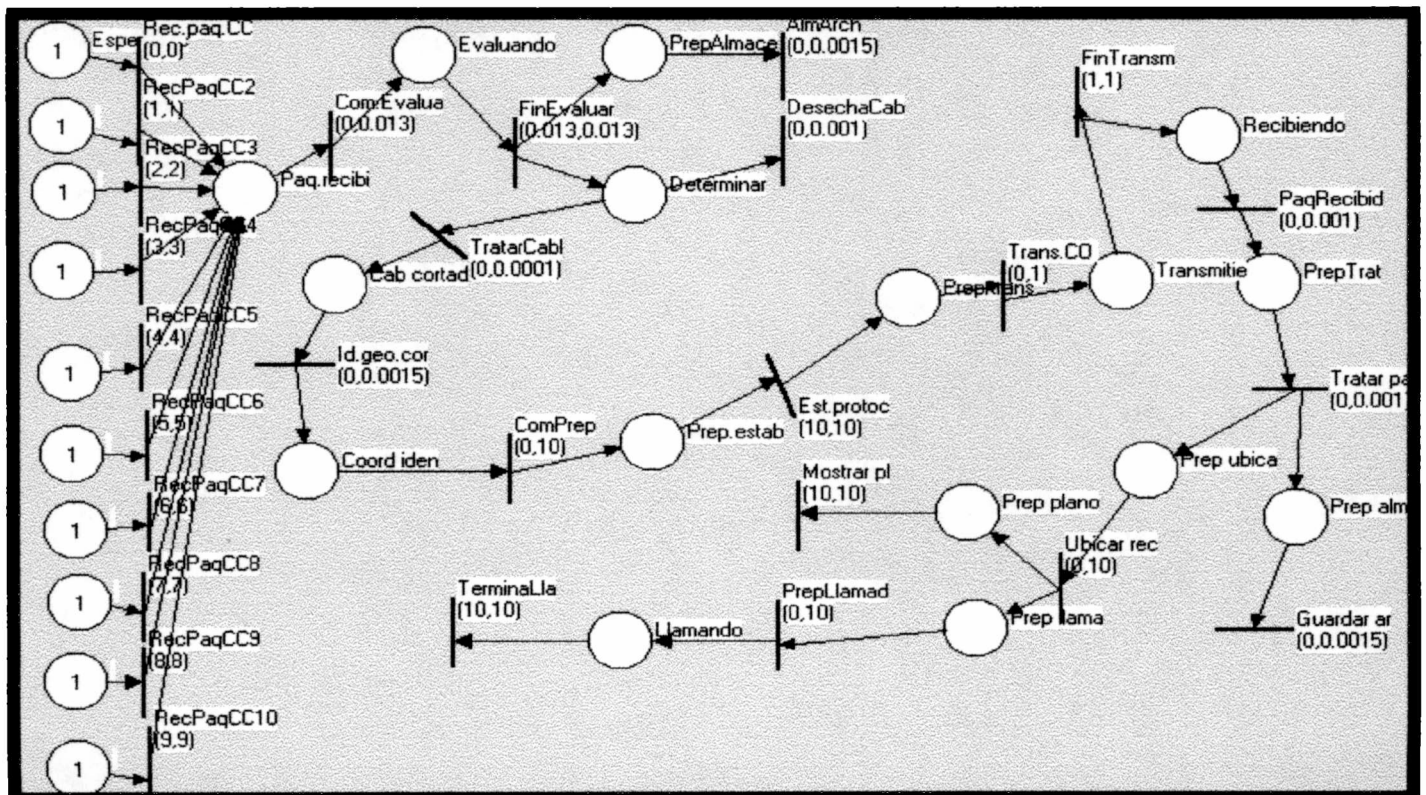
Arriba: Red de Petri con reintentos en la comunicación entre la Central y el Centro de Operaciones, comunicación máquina-Hombre entre el Centro de Operaciones y la Comisaria.

Abajo: Red de Petri con reintentos en ambas comunicaciones, comunicaciones máquina-máquina.





Arriba: Red de Petri con reintentos en ambas comunicaciones, y comunicación máquina-Hombre entre el Centro de Operaciones y la Comisaría.
Abajo: Muestra de modelización de la Red de Petri para la lectura de 10 cables.



4. Implementación del Prototipo

El prototipo desarrollado para la solución del problema planteado anteriormente, lo realizamos en Delphi 2.0, tomando como entrada una simulación de los datos que recibiremos del Sistema (S.L.T.) que mide la longitud de los cables telefónicos, realizado por los alumnos de Ingeniería Electrónica.

El prototipo consta de dos módulos, uno para cada Central y otro para el Centro de Operaciones, basándonos en un modelo Cliente/Servidor, donde cada Central es un cliente del centro de operaciones. Este, a su vez, es cliente del Agemap 3.5, en el cual se visualiza la ubicación del corte de la línea telefónica y en donde se encuentra la información de la comisaría correspondiente.

El módulo que va a estar ubicado en cada central monitorea periódicamente la información recibida desde el S.L.T., detectando si el cable monitoreado está cortado mediante una comparación de la longitud leída con la longitud real del cable. Si no se produjo un corte simplemente se almacena en un archivo histórico la información del cable monitoreado. Caso contrario se envía la información del cable cortado al módulo del Centro de Operaciones a través de un archivo compartido. En la realidad esto se realizará a través de una comunicación máquina-máquina.

El módulo que va a estar ubicado en el Centro de Operaciones, busca en el archivo compartido la información de algún cable cortado nuevo, el cual se envía al Agemap a través de otro archivo para su correspondiente visualización y obtención del teléfono de la comisaría de la zona a la cual pertenece el trayecto del cable cortado, activando un beep en modo de alarma.

Por otro lado, el Agemap cuando recibe un pedido de visualización, busca en su base de datos el trayecto correspondiente y la distancia del corte para poder ubicarlo geográficamente en el mapa. Además, dependiendo de la zona en donde se produjo el corte telefónico, se localiza el número de teléfono de la comisaría correspondiente enviando un mensaje del mismo por pantalla. En la realidad, además de aparecer el número en la pantalla de la computadora del Centro de Operaciones, se realizará una llamada a la comisaría correspondiente automáticamente e inmediatamente luego de ubicarse el corte. Esto se realizará a través de una comunicación máquina-máquina o máquina-hombre.

Esta fue la manera encontrada para implementar el prototipo del problema, ya que el Agemap 3.5 cuenta con una serie de limitaciones, ya que es una versión de demostración. Una de ellas es la imposibilidad de conocer las calles por las que pasa un trayecto. Esta opción está considerada, pero en esta versión está inhabilitada. Nuestra manera de resolverlo fue dividiendo el trazado del cable en trayectos, de los cuales conocemos las coordenadas de su punto inicial y punto final, y realizando la ubicación geográfica del corte a través de una búsqueda binaria en un segmento. Otra de las limitaciones es que la llamada telefónica a la comisaría no puede hacerse a través de este programa, por lo que esto se debería realizar desde un módulo Delphi.

Una de las ventajas de este G.I.S. es que se puede trabajar sobre el mapa y al mismo tiempo monitorear cables, o dado un trayecto de un cable recuperar información del mismo, como nombres de calles o alturas a través del manejo del mouse.

Otra opción del módulo del Centro de Operaciones es obtener reportes de la información histórica de los cortes de todas las centrales, una central en particular o un corte específico.

5. Conclusiones

Los problemas de Sistemas de Tiempo Real requieren extensiones a las técnicas clásicas de la Ingeniería de Software. En particular la metodología de K. Shumate, M. Keller, “Software Specification and Design” simplifica la derivación de la implementación de los módulos del prototipo.

El estudio de las respuestas en tiempo, requiere alguna herramienta de ejecución simulada. Las Redes de Petri extendidas demostraron ser amplias y útiles, ya que pudimos modelar las distintas situaciones planteadas con gran facilidad y claridad, y ejecutarlas obteniendo diferentes tiempos para luego realizar la evaluación de las respuestas del sistema.

En nuestra implementación el manejo de los datos (G.I.S.) se encuentra centralizado en el Centro de Operaciones, con lo cual no ofrece mayores problemas de administración.

Queda abierta la integración real del Sistema de Hardware desarrollado en la Facultad de Ingeniería, con un producto G.I.S. completo y los mecanismos de comunicación reales (modems).

6. Bibliografía

- Stan Aronoff, “Geographic Information Systems: An Management Perspective”, WDL Publications.
- K. Shumate, M. Keller, “Software Specification and Design”, John Wiley & Sons, 1992.
- A. Nielsen, “ADA in distributed real time systems”, Prentice Hall, 1992.
- María J. Abásolo, Marina L. Cantarella, Trabajo de Grado “Ambiente para la especificación y simulación de Sistemas de Tiempo Real con Redes de Petri extendidas”.

ÍNDICE

CAPITULO 1 – OBJETIVO DEL TRABAJO	1
1. MOTIVACIÓN Y OBJETIVO GENERAL	1
2. SOFTWARE A DESARROLLAR	1
CAPITULO 2 – SISTEMAS DISTRIBUIDOS DE TIEMPO REAL	2
1. CONCEPTOS SOBRE SISTEMAS DISTRIBUIDOS	2
1.1. Complejidad de los Sistemas Distribuidos	2
1.2. Sistemas Distribuidos de Tiempo Real	2
1.3. Tolerancia a fallas	3
1.3.1. Detección y corrección de errores	4
1.3.2. Procedimientos de recuperación	4
1.3.3. Balanceo de carga	4
2. SOPORTE PARA LOS SISTEMAS DISTRIBUIDOS	4
2.1. Especificación de Procesos y Soportes de Comunicación	4
2.1.1. Características de software de Sistemas Distribuidos	4
2.1.2. Modelos de comunicación de procesos	5
2.1.3. Mecanismos de comunicación remota	5
2.2. Entorno de soporte para Sistemas Distribuidos de Tiempo Real	6
2.2.1. Tiempo Real Ejecutivo	6
2.2.2. Servicios del Ejecutivo	7
CAPITULO 3 – ESPECIFICACIÓN Y DISEÑO DEL SOFTWARE	9
1. INTRODUCCIÓN	9
1.1. El desafío de la especificación y diseño del Software	9
2. INGENIERÍA DE SISTEMAS Y EL PROCESO DE DESARROLLO DEL SISTEMA	10
2.1. Desarrollo de grandes sistemas	10
3. DOCUMENTACIÓN	11
CAPITULO 4 – REDES DE PETRI.....	18
1. INTRODUCCIÓN	18
2. REDES DE PETRI.....	18
2.1. Definiciones básicas	18
2.2. Grafo en la Red de Petri	18
2.3. Reglas de ejecución en una Red de Petri marcada	19
3. EXTENSIONES DE LAS REDES DE PETRI PARA MODELAR SISTEMAS DE TIEMPO REAL	19
3.1. Extensiones de tiempo	19
3.2. Extensiones con respecto a los arcos	20
3.3. Definición formal de la red de Petri extendida	21
4. EJECUCIÓN DE LAS REDES DE PETRI EXTENDIDAS	21
4.1. Reglas de ejecución	21
4.2. Trama o secuencia de la ejecución	22
CAPITULO 5 – SISTEMAS DE INFORMACIÓN GEOGRÁFICA.....	23
1. ¿QUÉ ES UN SISTEMA DE INFORMACIÓN GEOGRÁFICA?	23
2. SUBSISTEMAS DE UN G.I.S.....	24
3. PRINCIPALES FORMAS DE USO Y APLICACIONES DE UN G.I.S.	24
CAPITULO 6 – AGEMAP 3.5	26
1. INTRODUCCIÓN	26
2. MAPAS	26
3. CONSULTA DE ELEMENTOS DE UNA BASE DE DATOS	26
4. DISTANCIAS	26

5. BARRIOS.....	26
6. BASES DE DATOS	27
7. GEOTIPOS.....	27
8. GEOCODIZADOR.....	31
8.1. <i>Modo automático</i>	31
8.2. <i>Modo interactivo</i>	31
9. COMANDOS Y FUNCIONES GEOGRÁFICAS	32
10. ZONAS.....	33
11. TRAYECTOS	33
12. MANEJO DE DDEML	34
 CAPITULO 7 – DESARROLLO DE LA APLICACIÓN.....	35
1. ESPECIFICACIÓN Y DISEÑO DE SISTEMAS DE SOFTWARE DE TIEMPO REAL.....	35
1.1. <i>Ingeniería del Sistema</i>	35
1.1.1. Análisis de requerimientos del Sistema	35
1.1.2. Diseño del Sistema.....	47
1.2. <i>Ingeniería del Software</i>	53
1.2.1. Análisis de requerimientos del Software	53
1.2.2. Diseño top level	71
1.2.3. Diseño detallado	82
2. ANÁLISIS DE TIEMPOS	92
3. ALGUNAS REDES DE PETRI IMPLEMENTADAS	98
4. IMPLEMENTACIÓN DEL PROTOTIPO.....	102
5. CONCLUSIONES.....	103
6. BIBLIOGRAFÍA	104